



TUGAS AKHIR - SM 141501

**APLIKASI DATA MINING PADA PENILAIAN KREDIT
MENGUNAKAN ALGORITMA *ROUGH SET***

**ANISA SOFIA
NRP 1210 100 024**

**Dosen Pembimbing
Dr. Imam Mukhlash, S.Si., M.T.**

**JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT-SM 091332

**APPLICATION OF DATA MINING FOR CREDIT SCORING
USING ROUGH SET ALGORITHM**

**ANISA SOFIA
NRP 1210 100 024**

**Supervisor
Dr. Imam Mukhlash, S.Si., M.T.**

**DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2015**

LEMBAR PENGESAHAN

APLIKASI DATA MINING PADA PENILAIAN KREDIT MENGGUNAKAN ALGORITMA ROUGH SET

APPLICATION OF DATA MINING FOR CREDIT SCORING USING ROUGH SET ALGORITHM

TUGAS AKHIR


Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

ANISA SOFIA

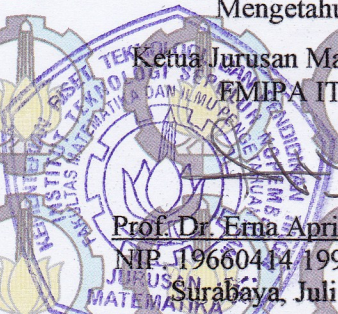
NRP. 1210100 024

Menyetujui,
Dosen Pembimbing,


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831199403 1 003

Mengetahui,

Ketua Jurusan Matematika
FMIPA ITS


Prof. Dr. Erna Apriliani, M.Si
NIP. 19660414 199102 2 001
Surabaya, Juli 2015

APLIKASI DATA MINING PADA PENILAIAN KREDIT MENGUNAKAN ALGORITMA ROUGH SET

Nama Mahasiswa : ANISA SOFIA
NRP : 1210 100 024
Jurusan : Matematika
Dosen Pembimbing : Dr. Imam Mukhlash, S.Si., M.T.

Abstrak

Kredit merupakan salah satu mekanisme pembayaran yang umum di masyarakat. Banyaknya permohonan kredit menuntut kreditor harus mampu mengevaluasi permohonan kredit dengan objektif, akurat, dan konsisten. Oleh karena itu diperlukan credit scoring untuk membantu kreditor mengevaluasi permohonan kredit. Permasalahan yang dibahas dalam Tugas Akhir ini adalah mengetahui potensi keberhasilan seorang calon nasabah kredit menggunakan algoritma klasifikasi berbasis rough set dengan data historis nasabah kredit yang sudah ada. Parameter-parameter yang digunakan adalah suku bunga, besar kredit, kolektibilitas, jangka waktu, jenis usaha, produk dan omzet nasabah. Dari hasil penelitian, didapatkan sebuah model rough set yang mampu memprediksi keberhasilan calon nasabah dalam pengambilan kredit. Dengan tingkat akurasi sebesar 82,4%. Dengan adanya aplikasi ini, seorang pengambil keputusan dapat mengetahui potensi calon nasabah dan mencegah terjadinya kredit macet.

Kata Kunci : Data Mining, Credit Scoring, Klasifikasi, Rough Set

APPLICATION OF DATA MINING FOR CREDIT SCORING USING ROUGH SET ALGORITHM

Student Name : ANISA SOFIA
NRP : 1210 100 024
Departments : Mathematics
Supervisors : Dr. Imam Mukhlash, S.Si., M.T.

Abstract

Credit is one of payment mechanism in society. Because there are many credit application, creditors needs to evaluate the credit application with objectively, accurately, and consistenlyt. Therefore, it is needed credit scoring to help creditors evaluate a credit application. The problem discussed is knowing the potential success of a prospective customer credit through classification algorithm based on rough sets with existing credit historical data. The parameters used are: plafond, rate, term, product, turnover, type of business and collectibility. From the research, obtained a model of rough set with an accuracy level of 82,4%. With this application, a decision maker can determine the potential prospective customer credit and prevent bad debts.

Keywords — Data Mining, Credit Scoring, Classification, Rough Set

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan limpahan nikmat, karunia, serta hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“APLIKASI DATA MINING PADA PENILAIAN KREDIT MENGUNAKAN ALGORITMA ROUGH SET”

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Studi S-1 pada Jurusan Matematika Fakultas MIPA Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik dan lancar berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis bermaksud menyampaikan ucapan terima kasih kepada:

1. Dr. Imam Muklash S.Si, MT selaku dosen pembimbing.
2. Dr. Erna Apriliani, M.Si selaku Ketua Jurusan Matematika.
3. Dian Winda Setyawati, S.Si, M.Si selaku Dosen Wali.
4. Dr. Budi Setiyono, S.Si, MT. Dr. Dwi Ratna Sulistyaningrum, S.Si, MT. Moh. Iqbal, S.Si, M.Si. Dr. Chairul Imron, MI.Komp selaku dosen penguji.
5. Dr. Chairul Imron, MI.Komp selaku koordinator Tugas Akhir.
6. Seluruh jajaran dosen dan staf jurusan Matematika ITS.
7. Seluruh teman-teman mahasiswa jurusan Matematika ITS.

special thank's to

Selama proses pembuatan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan dan dukungannya terhadap penulis. Rasanya tidak akan puas hati penulis jika belum mengucapkan terima kasih secara khusus dan apresiasi yang sebesar-besarnya kepada:

1. Ibu Ir.Sugistiana Dwi Lusilowati dan Bapak Ir.Suhadak. Terima kasih atas do'a, perhatian, serta pengorbanan yang tak habis-habisnya tcurahkan kepada anakmu ini.
2. Adek Nabilah Amirah. Terima kasih semangatnya.
3. Gede Eka Praselia. Terima kasih sudah sangat sabar memberi nasehat, semangat serta doa.
4. Teman-teman jurusan Matematika seperjuangan wisuda 112.
5. Semua pihak yang telah mendukung pengerjaan Tugas Akhir ini yang tidak bisa disebutkan satu per satu.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat memberikan manfaat pada semua pihak yang berkepentingan.

Surabaya, Agustus 2015

Penulis

DAFTAR ISI

| | Halaman |
|--|---------|
| HALAMAN JUDUL | i |
| HALAMAN PENGESAHAN | v |
| ABSTRAK | vii |
| ABSTRACT | ix |
| KATA PENGANTAR | xi |
| DAFTAR ISI | xiii |
| DAFTAR TABEL | xv |
| DAFTAR GAMBAR | xvii |
| BAB I. PENDAHULUAN | |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan | 3 |
| 1.5 Manfaat | 3 |
| 1.6 Sistematika Penulisan | 4 |
| BAB II. TINJAUAN PUSTAKA | |
| 2.1 Penelitian Sebelumnya..... | 5 |
| 2.2 Penilaian Kredit (<i>Credit Scoring</i>) | 6 |
| 2.3 Data Mining | 8 |
| 2.3.1. Metode Data Mining..... | 9 |
| 2.3.2. Beberapa faktor Pendukung Adanya Data Mining | 10 |
| 2.3.3. Model Dalam Data Mining | 11 |
| 2.3.4. Tahapan Data Mining | 11 |
| 2.4 <i>Klasifikasi</i> | 13 |
| 2.5 <i>Decision Tree</i> | 14 |
| 2.6 <i>Decision Rule</i> | 14 |
| 2.6.1. Algoritma <i>Decision Rule</i> | 14 |
| 2.7 <i>MySQL</i> | 15 |
| 2.7.1. Keunggulan <i>MySQL</i> | 16 |
| 2.8 <i>Rough Set</i> | 17 |

| | |
|---|----|
| 2.8.1. Istilah dalam <i>Rough Set</i> | 18 |
| 2.8.2. Algoritma Klasifikasi Berbasis <i>Rough Set</i> | 19 |
| 2.8.3. Contoh Klasifikasi Berbasis <i>Rough Set</i> | 19 |
| 2.9 Alur Proses Permohonan Kredit Bank “X” di Surabaya..... | 23 |
| BAB III. METODE PENELITIAN | |
| 3.1 Langkah Pengerjaan..... | 27 |
| 1. Studi Pustaka..... | 27 |
| 2. Pengumpulan Data..... | 27 |
| 3. Pengembangan Perangkat Lunak..... | 27 |
| 4. Mengevaluasi Hasil Kerja dan Kesimpulan.... | 28 |
| 3.2 Diagram Alir Penelitian | 29 |
| BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM | |
| 4.1 Analisis Sistem..... | 31 |
| 4.1.1 Deskripsi Perangkat Lunak | 31 |
| 4.1.2 Pemodelan Sistem | 32 |
| 4.1.3 Identifikasi Permasalahan | 33 |
| 4.1.4 Penentuan <i>Range</i> | 34 |
| 4.1.5 Pembentukan Data Kategorial | 34 |
| 4.2 Perancangan Sistem | 35 |
| 4.2.1 Perancangan Umum | 35 |
| 4.2.2 Perancangan Data..... | 37 |
| 4.2.2.1 <i>Design Database</i> | 38 |
| 4.2.3 Perancangan Proses..... | 41 |
| 4.3 Implementasi Sistem..... | 48 |
| 4.3.1 Lingkungan Implementasi..... | 48 |
| 4.3.1.1 Lingkungan Perangkat Keras | 48 |
| 4.3.1.2 Lingkungan Perangkat Lunak | 48 |
| 4.3.2 Implementasi Program | 48 |
| 4.3.2.1 Implementasi Konversi Kategorial | 49 |
| 4.3.2.2 Implementasi Matriks <i>Discernibility</i> | 49 |
| 4.3.2.1 Implementasi <i>Rules</i> | 49 |
| 4.3.2.1 Implementasi Uji Data | 50 |

BAB V. UJI COBA DAN PEMBAHASAN

| | |
|--|----|
| 5.1 Ruang Lingkup Uji Coba..... | 51 |
| 5.2 Uji Tampilan Awal | 52 |
| 5.3 Uji Halaman Penampilan Data | 52 |
| 5.4 Uji halaman data yang telah diubah kedalam bentuk kategorial..... | 53 |
| 5.5 Uji halaman data training..... | 54 |
| 5.6 Halaman Data Uji | 55 |
| 5.7 Halaman Uji Calon Nasabah Kredit | 55 |
| 5.8 Halaman Uji Coba..... | 57 |

BAB VI. PENUTUP

| | |
|----------------------|----|
| 6.1 Kesimpulan | 61 |
| 6.2 Saran | 61 |

DAFTAR PUSTAKA

LAMPIRAN.....

DAFTAR TABEL

| | Halaman |
|--|---------|
| Tabel 4.1 <i>Range</i> Untuk Setiap Parameter | 34 |
| Tabel 4.2 Nilai Kategori | 35 |
| Tabel 5.1 Ruang Lingkup Uji Coba | 51 |
| Tabel 5.2 Data Uji <i>k-fold cross validation</i> | 57 |
| Tabel 5.3 Nilai Akurasi <i>k-fold cross validation</i> | 58 |

[halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | Halaman |
|--|---------|
| Gambar 2.1 Tahap-Tahap Data Mining | 12 |
| Gambar 2.2 Contoh Data Nasabah Kredit..... | 19 |
| Gambar 2.3 Contoh Data Kategorial..... | 22 |
| Gambar 2.4 Matriks <i>Discernibility</i> | 22 |
| Gambar 2.5 Alur Proses Permohonan Kredit..... | 25 |
| Gambar 3.1 Diagram Alir Penelitian | 29 |
| Gambar 3.2 Diagram Alir Penelitian (Lanjutan)..... | 30 |
| Gambar 4.1 Diagram <i>Use Case</i> Keseluruhan | 32 |
| Gambar 4.2 <i>State Diagram</i> Aplikasi <i>Credit Scoring</i> | 36 |
| Gambar 4.3 <i>Sequence Diagram</i> Aplikasi <i>Credit Scoring</i> .. | 37 |
| Gambar 4.4 Data Nasabah Kredit pada <i>Microsoft Excel</i> .. | 38 |
| Gambar 4.5 <i>Design CDM Database</i> | 39 |
| Gambar 4.6 <i>Design PDM Database</i> | 40 |
| Gambar 4.7 Implementasi <i>Database</i> di <i>MySQL</i> | 40 |
| Gambar 4.8 Rancangan Umum Program..... | 41 |
| Gambar 4.9 Proses Klasifikasi Berbasis <i>rough Set</i> | 42 |
| Gambar 4.10 Proses pembentukan data kategorial..... | 42 |
| Gambar 4.11 Langkah-langkah proses klasifikasi..... | 43 |
| Gambar 5.1 Halaman Login..... | 52 |
| Gambar 5.2 Halaman Penampilan Data..... | 53 |
| Gambar 5.3 Halaman Penampilan Data Kategorial | 54 |
| Gambar 5.4 Halaman Penampilan Uji Data Training | 54 |
| Gambar 5.5 Halaman Penampilan Uji | 55 |
| Gambar 5.6 Halaman Contoh Prediksi Calon Nasabah Macet..... | 56 |
| Gambar 5.7 Halaman Contoh Prediksi Calon Nasabah Lancar..... | 57 |
| Gambar 5.8 Halaman Uji <i>K-fold Cross Validation</i> | 58 |
| Gambar 5.9 Halaman Proses Uji Coba..... | 59 |
| Gambar 5.10 Halaman Nilai Akurasi..... | 59 |

BAB I

PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan, dan manfaat, serta sistematis penulisan dalam Tugas Akhir ini.

1.1. Latar Belakang

Perbankan merupakan lembaga yang sangat vital bagi perekonomian Indonesia[4]. Salah satunya dikarenakan kemampuan bank untuk memberikan pinjaman bagi nasabahnya. Dengan berkembangnya zaman, perkembangan industri semakin berkembang pesat pula. Hal ini membuat penyediaan barang kebutuhan masyarakat sangat melimpah. Kebanyakan masyarakat kita mudah tertarik untuk mengkonsumsi barang dengan banyak pilihan yang ada, sesuai dengan kebutuhan masing-masing. Bagi masyarakat menengah keatas hal ini tidak memberikan efek yang berarti, akan tetapi tidak bagi masyarakat menengah kebawah yang penghasilannya hanya cukup untuk kehidupan sehari-hari. Masyarakat kalangan menengah ke bawah cenderung membeli barang, kendaraan, dan rumah dengan cara kredit.

Banyak Bank yang menyediakan fasilitas kredit bagi para nasabahnya. Pengajuan kredit bisa diajukan oleh para pemohon dengan memenuhi syarat-syarat yang ditentukan oleh Bank terkait. Dalam kegiatan pengkreditan sering terjadi masalah kredit macet atau kredit bermasalah sehingga menyebabkan *defisit* yang disebabkan oleh gagalnya pengembalian sebagian pinjaman yang diberikan kepada para peminjam. Untuk mengatasi masalah ini, salah satunya dengan mengidentifikasi dan memprediksi pemohon kredit dengan baik sebelum memberikan pinjaman dengan cara memperhatikan data historis peminjam secara umum.

Dengan melakukan penilaian kredit (*credit scoring*), pemohon kredit dapat diklasifikasikan menjadi dua kategori berdasarkan kemungkinan kemampuan pembayaran mereka yaitu pemohon kredit yang baik dan cenderung membayar kewajiban

cicilan setelah menerima kredit dan pemohon yang buruk akan ditolak karena besar kemungkinan terjadi gagal membayar kewajiban cicilan. Salah satu metode *credit scoring* yang digunakan oleh Bank X selama ini yaitu dengan melihat dokumen legal, jurnal transaksi, laporan keuangan 3 tahun terakhir dan data jaminan (SHM/SHGB, IMB, PBB) yang dimiliki oleh pemohon kredit sehingga komite kredit di “Bank X” tersebut bisa mengambil keputusan diterima atau ditolak pengusulan kredit tersebut.

Banyaknya kegiatan yang dilakukan secara terkomputerisasi menyebabkan penyimpanan data menjadi semakin mudah. Namun data yang dibiarkan menumpuk tanpa dianalisis lebih dalam mengakibatkan data tersebut tidak mempunyai nilai guna lebih di masa mendatang. Persoalan tersebut merupakan salah satu persoalan yang dapat diatasi oleh data mining. Didalam data mining terdapat beberapa metode: klasifikasi, *association rules*, *sequence pattern*, dan lain-lain.

Klasifikasi adalah suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan[6]. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi.

Secara umum, teori *rough set* adalah himpunan kasar yang nilai atributnya sama tetapi mungkin punya keputusan yang berbeda[7]. *Rough set* telah digunakan dalam banyak aplikasi seperti *medicine*, *pharmacology*, *business*, *banking*, *engineering design*, *image processing* dan *decision analysis*.

Tugas Akhir ini, bertujuan untuk mengetahui karakteristik calon nasabah kredit apakah layak diberikan kredit atau tidak dengan data historis nasabah kredit yang sudah ada.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, permasalahan dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana memprediksi karakteristik nasabah kredit yang beresiko kredit baik dan nasabah kredit yang beresiko kredit buruk dengan melakukan pengenalan pola data historis untuk digolongkan kedalam kredit lancar dan kredit bermasalah (macet)
2. Bagaimana mengaplikasikan algoritma *rough set* untuk prediksi penilaian kredit dari calon nasabah kredit “Bank X” di Surabaya

1.3. Batasan Masalah

Pembahasan pada Tugas Akhir ini dibatasi pada beberapa hal berikut:

1. Sistem penilaian kredit ini hanya mengetahui apakah seorang calon nasabah kredit akan mengalami kredit lancar/macet.
2. Parameter yang digunakan suku bunga, omzet, produk, jenis usaha, plafond, kolektibilitas dan jangka waktu.
3. Aplikasi penilaian kredit yang dibangun menggunakan perangkat lunak Netbeans IDE dengan Pemrograman Java dan MySQL sebagai DBMS-nya

1.4. Tujuan

Tujuan dari Tugas Akhir ini adalah:

1. Mengembangkan aplikasi data mining dengan algoritma klasifikasi berbasis *rough set*
2. Mendapatkan karakteristik calon nasabah kredit dari data historis nasabah yang ada.
3. Mengetahui hasil prediksi dari model algoritma klasifikasi berbasis *rough set*.

1.5. Manfaat

Dengan dibangunnya perangkat lunak evaluasi kredit ini diharapkan dapat membantu analisis kredit pada Bank X dalam melakukan penilaian kredit agar mengurangi resiko kredit macet.

1.6. Sistematika Penulisan

Tugas Akhir ini disusun berdasarkan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan pada Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Menjelaskan dasar teori yang digunakan penulis dalam mengerjakan Tugas Akhir. Berisi tentang pengertian dari *credit scoring*, *rough set*, algoritma klasifikasi *rough set*, serta alur proses permohonan kredit di Bank "X".

BAB III METODOLOGI TUGAS AKHIR

Menjelaskan alur kerja dan metode yang digunakan penulis dalam mengerjakan Tugas Akhir ini.

BAB IV ANALISIS DAN PEMBAHASAN

Menyajikan tentang analisa data dan pembahasan dalam pembentukan algoritma *rough set*

BAB V UJI COBA DAN PEMBAHASAN

Pada tahap ini, dilakukan uji coba pada aplikasi yang telah dibuat dan mengetahui akurasi data yang diuji

BAB VI PENUTUP

Berisi kesimpulan dari hasil pembahasan dalam Tugas Akhir ini dan saran penulis.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas teori-teori yang terkait dengan permasalahan dalam Tugas Akhir ini. Seperti pengertian dari *credit scoring*, *data mining*, *rough set*, algoritma klasifikasi *rough set*, serta alur proses permohonan kredit Bank “X” di Surabaya.

2.1. Penelitian Sebelumnya

Penelitian tentang permasalahan penilaian kredit telah banyak dilakukan diantaranya yang pertama adalah Aprianti, Winda dan Mukhlash, Imam, ICoDSE 2014 [2] “*The Application of Rough Set and Fuzzy Rough Set Based Algorithm to Classify Incomplete Meteorological Data*”. Pada penelitian ini penulis menggunakan 2 pendekatan untuk mengatasi data yang ada, 2 pendekatan itu adalah *Algoritma Rough Set* dan *Algoritma Fuzzy Rough Set*. Untuk menguji kedua algoritma tersebut, penulis menggunakan data meteorologi untuk mengklasifikasikan cuaca hujan atau cuaca kering. Kesimpulan dari penelitian ini menunjukkan bahwa algoritma *rough set* lebih efisien daripada pendekatan *algoritma fuzzy rough set*.

Yang kedua adalah Wahyuningtyas, Grizelda. 2014[1] “Aplikasi Data Mining untuk Penilaian Kredit Menggunakan Metode Pohon Keputusan Fuzzy”. Pada penelitian ini penulis menggunakan *fuzzy decision tree* yang merupakan salah satu metode data mining yang berisi gabungan antara *decision tree* dengan *logika fuzzy*. Penerapan metode tersebut bertujuan untuk mempresentasikan suatu kondisi yang tidak pasti dan sangat kompleks. Pembentukan *fuzzy decision tree* menggunakan teknik *fuzzy* secara kasar (*fuzzy rough technique*) dengan cara mencari nilai bawah dan *significance level* untuk setiap factor yang dianalisis. Untuk menguji metode tersebut, penulis menggunakan data nasabah kredit micro dari BRI cabang Kediri. Dari hasil

penelitian, didapatkan model pohon keputusan *fuzzy* dengan tingkat akurasi sebesar 83%[1].

Penelitian yang ketiga adalah Claudia. C, Ciptohartono. 2014[4] “*Algoritma Klasifikasi Naïve Bayes untuk Menilai Kelayakan Kredit*”. Pada penelitian ini penulis menggunakan *Model Naïve Bayes*. Dalam model ini diasumsikan bahwa hubungan antar fitur atau atribut independen sehingga menjadikannya sederhana dan efisien. Untuk menguji metode tersebut, penulis menggunakan dataset konsumen dari perusahaan BCA Finance Jakarta 2013. Kesimpulan dari penelitian ini yaitu penerapan *Naïve Bayes* diharapkan dapat menilai akurasi dengan biaya terendah dalam menentukan nilai kelayakan kredit.

Penelitian yang keempat adalah Y. Ping and L Yongheng, 2011[5] “*Neighborhood rough set and SVM based hybrid credit scoring classifie*”. Penelitian ini bertujuan untuk meningkatkan akurasi *credit scoring* dengan membangun sebuah model *credit scoring* berbasis *hybrid SVM* sesuai inputan atribut pemohon. Dua masalah dihadapi yaitu memilih input atribut dan menentukan parameter kernel terbaik. Untuk menyelesaikan hal ini tiga strategi: (1) menggunakan *neighborhood rough set* untuk memilih masukan atribut, (2) menggunakan pencarian *Grid Search* untuk optimal parameter kernel RBF, (3) menggunakan masukan atribut *hybrid* yang optimal dan model parameter untuk memecahkan masalah *credit scoring* dengan *10-fold cross validation*, (4) membandingkan tingkat akurasi dengan metode lain. Hasil eksperimen menunjukkan bahwa *neighborhood rough set* dan *SVM based hybrid classifier* paling bagus dengan *hybrid* klasifikasi lainnya yaitu *linear discriminant analysis*, *logistic regression* dan *neural networks*. Nilai akurasi 76,60%[5].

2.2. Penilaian Kredit (*Credit Scoring*)

Kredit merupakan salah satu mekanisme pembayaran yang sangat umum di masyarakat. Fungsi pokok kredit yaitu memenuhi pelayanan terhadap kebutuhan masyarakat dalam rangka memperlancar perdagangan, produksi dan jasa-jasa

bahkan konsumsi yang semuanya ditujukan untuk meningkatkan kesejahteraan manusia.

Pihak pemberi dan penerima kredit harus bekerjasama untuk mencapai tujuan yang diinginkan. Pihak penerima kredit mempunyai tujuan mendapatkan sumber pembiayaan yang mudah dan cepat. Pihak pemberi kredit mempunyai tujuan mendapatkan keuntungan dari bunga yang dibayarkan oleh nasabah kredit. Disamping keuntungan yang dapat diperoleh, pemberi kredit juga harus menanggung resiko ketidaklancaran pembayaran oleh nasabah. Pihak kreditor harus mengevaluasi setiap pemohon kredit sebelum keputusan penerimaan atau penolakan diambil. Banyaknya permohonan kredit menuntut pemberi kredit harus mampu mengevaluasi pemohon kredit dengan objektif, akurat, dan konsisten. Evaluasi tersebut dapat dibantu dengan *credit scoring*.

Credit scoring adalah *tools* yang melibatkan penggunaan model statistik untuk mengevaluasi seluruh informasi yang tersedia dengan objektif dalam pengambilan keputusan kredit[3]. *Credit scoring* adalah teknik yang paling umum digunakan untuk mengevaluasi kelayakan kredit dari pemohon kredit dengan segala atribut yang mereka miliki. Manfaat yang dapat diperoleh dari penerapan *credit scoring* adalah peningkatan kecepatan dan konsistensi proses aplikasi pinjaman dan memungkinkan otomatisasi proses peminjaman.

Model penilaian kredit dibangun dengan menggunakan sampel kredit masa lalu dalam jumlah yang besar. Sampel tersebut dibagi kedalam dua kelas yaitu kredit yang baik (pembayaran dilakukan tepat waktu) dan kredit yang bermasalah (pembayaran dilakukan tidak tepat waktu atau tidak dapat melakukan pembayaran). Berdasarkan pola masa lalu, kombinasi karakteristik peminjam yang membedakan peminjam yang baik dan yang buruk menghasilkan nilai sebagai estimasi resiko dari tiap peminjam baru. Sebelum para nasabah kredit menerima pinjaman kredit, mereka harus melalui aturan-aturan yang telah disediakan oleh bank yang terkait.

2.3. Data Mining [6]

Data tidak dapat dipisahkan dari kehidupan sehari-hari. Suatu organisasi dapat dibanjiri dengan berbagai macam data. Sangatlah tidak berguna data yang terkumpul dan tersimpan jika tidak dimanfaatkan. Permasalahan yang muncul adalah bagaimana memproses data-data tersebut sehingga dapat menampilkan suatu pengetahuan (knowledge) yang berguna bagi manajemen dalam mengambil keputusan.

Secara umum data mining dapat didefinisikan sebagai berikut:

1. Proses penemuan pola yang menarik dari data yang tersimpan dalam jumlah besar. Merupakan evolusi alami dari teknologi database, dan merupakan metode yang paling banyak dibutuhkan, dengan aplikasi yang sangat luas.
2. Ekstraksi dari suatu informasi yang berguna atau menarik (non-trivial, implisit, sebelumnya belum diketahui, potensial kegunaannya) pola atau pengetahuan dari data yang disimpan dalam jumlah besar.
3. Eksplorasi dari analisa secara otomatis atau semiotomatis terhadap data-data dalam jumlah besar untuk mencari pola dan aturan yang berarti.

Data mining dapat didefinisikan juga sebagai suatu proses yang menggunakan berbagai perangkat analisis data untuk menemukan pola dan relasi data agar dapat digunakan untuk membuat prediksi yang tepat.

Data mining membantu perusahaan untuk mendapatkan pola dari data-data yang tersimpan di dalam basis data perusahaan. Pengetahuan yang diperoleh tersebut akan menjadi pedoman dalam mengambil tindakan-tindakan bisnis sebagai upaya pemeliharaan dan peningkatan tingkat kompetitif bisnis perusahaan.

2.3.1. Metode Data Mining

Metode data mining dapat diklasifikasikan berdasarkan fungsi yang dilakukan atau berdasarkan jenis aplikasi yang menggunakannya:

1. Klasifikasi (*supervised learning*)

Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi. Teknik ini menggunakan *supervised induction*, yang memanfaatkan kumpulan pengujian dari *record* yang terklasifikasi untuk menentukan kelas-kelas tambahan.

2. *Clustering* (*unsupervised learning*)

Digunakan untuk menganalisis pengelompokkan berbeda terhadap data, mirip dengan klasifikasi, namun pengelompokkan belum didefinisikan sebelum dijalankannya tool data mining. Biasanya menggunakan metode *neural network* atau statistik. *Clustering* membagi item menjadi kelompok-kelompok berdasarkan yang ditemukan *tool* data mining. Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar *cluster*. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang multidimensi.

3. *Association Rules*

Digunakan untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana link asosiasi muncul pada setiap kejadian. Contoh dari aturan assosiatif dari analisa pembelian disuatu pasar swalayan adalah bisa diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan

susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* yaitu prosentasi kombinasi atribut tersebut dalam basis data dan *confidence* yaitu kuatnya hubungan antar atribut dalam aturan asosiatif. Motivasi awal pencarian *association rule* berasal dari keinginan untuk menganalisa data transaksi supermarket, ditinjau dari perilaku *customer* dalam membeli produk. Association rule ini menjelaskan seberapa sering suatu produk dibeli secara bersamaan.

2.3.2. Beberapa Faktor Pendukung Adanya Data Mining

Beberapa faktor yang mendukung perlunya dilakukan data mining adalah:

1. Data telah mencapai jumlah dan ukuran yang sangat besar
 Hasil proses data mining merupakan suatu informasi yang akan mendasari tindakan tertentu sehingga tingkat kebenaran informasi tersebut menjadi sangat signifikan, dan makin besar serta makin banyak data yang digunakan maka akan semakin valid hasilnya.
2. Telah dilakukan proses data *warehousing*.
 Untuk mencapai hasil yang memuaskan, maka sumber data yang digunakan dalam proses data mining seringkali merupakan data gabungan dari banyak departemen, daerah operasi bahkan dari sumber-sumber lain seperti data kependudukan.
3. Kemampuan komputasi yang semakin terjangkau.
 Pada dasarnya proses data mining melakukan banyak akses terhadap data yang sangat besar. Selain itu juga melakukan proses komputasi yang membutuhkan sumber daya sangat besar.
4. Persaingan bisnis yang semakin ketat.
 Tekanan persaingan bisnis yang semakin ketat mendorong perusahaan-perusahaan untuk selalu berinovasi agar mampu

meningkatkan daya saingnya dipasar global. Beberapa tren yang berkembang saat ini adalah:

- a. Setiap bisnis adalah bisnis pelayanan
- b. Adanya fenomena kustomisasi produk oleh masyarakat.
- c. Informasi adalah produk

2.3.3. Model Dalam Data Mining

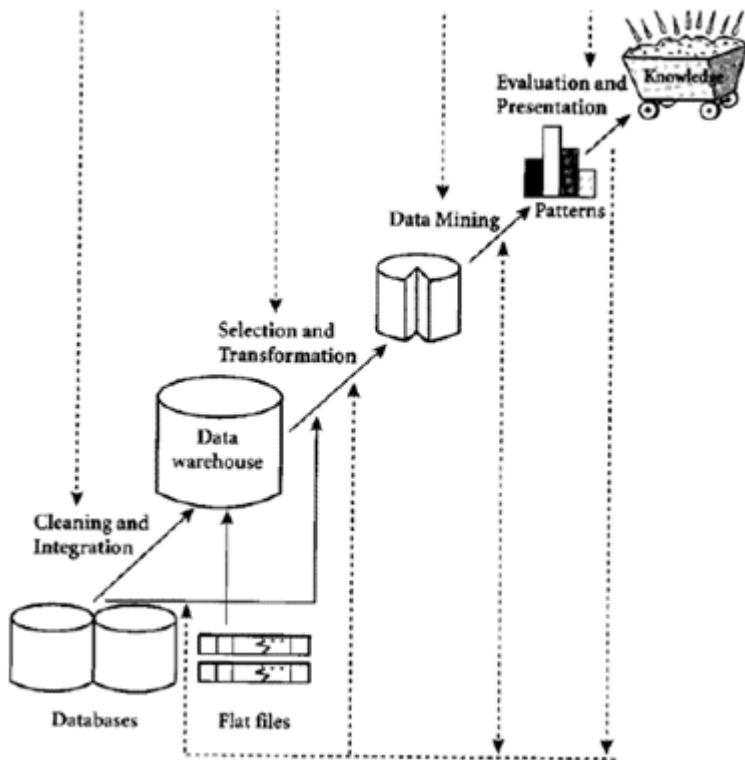
Terdapat dua tipe atau mode operasi yang bisa digunakan untuk mencari informasi yang dibutuhkan *user* lewat proses data mining, yaitu model verifikasi dan *knowledge discovery*.

Model verifikasi menggunakan pendekatan *top down* dengan mengambil hipotesa dari *user* dan memeriksa validitasnya dengan data sehingga bisa dibuktikan kebenaran hipotesa tersebut.

Model *knowledge discovery* menggunakan pendekatan *bottom up* untuk mendapatkan informasi yang sebelumnya tidak diketahui.

2.3.4. Tahapan Data Mining

Salah satu tuntutan dari data mining ketika diterapkan pada data berskala besar adalah diperlukan metodologi sistematis tidak hanya ketika melakukan analisa saja tetapi juga ketika mempersiapkan data dan juga melakukan interpretasi dari hasilnya sehingga dapat menjadi aksi ataupun keputusan yang bermanfaat. Data mining seharusnya dipahami sebagai suatu proses, yang memiliki tahapan-tahapan tertentu dan juga ada umpan balik dari setiap tahapan ke tahapan sebelumnya. Pada umumnya proses data mining berjalan interaktif karena tidak jarang hasil data mining pada awalnya tidak sesuai dengan harapan analisnya sehingga perlu dilakukan desain ulang prosesnya.



Gambar 2.1. Tahap-tahap Data mining[6]

Sebagai suatu rangkaian proses, data mining dapat dibagi menjadi beberapa tahap yang diilustrasikan di Gambar 2.1. Tahap-tahap tersebut bersifat interaktif di mana pemakai terlibat langsung atau dengan perantara *knowledge base*.

1. Pembersihan data, Digunakan untuk membuang data yang tidak konsisten dan noise.
2. Intergrasi Data, Data yang diperlukan untuk data mining tidak hanya berasal dari satu database tetapi juga berasal dari beberapa *database* atau file teks. Hasil integrasi data sering

diwujudkan dalam sebuah data *warehouse* karena dengan data *warehouse*, data dikonsolidasikan dengan struktur khusus yang efisien. Selain itu data *warehouse* juga memungkinkan tipe analisa seperti OLAP.

3. Transformasi data, Transformasi dan pemilihan data ini untuk menentukan kualitas dari hasil data mining, sehingga data diubah menjadi bentuk sesuai untuk di-Mining.

4. Aplikasi Teknik Data Mining, hanya merupakan salah satu bagian dari proses data mining.

5. Evaluasi pola yang ditemukan, Dalam tahap ini hasil dari teknik data mining berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai.

6. Presentasi Pengetahuan, Presentasi pola yang ditemukan untuk menghasilkan aksi tahap terakhir dari proses data mining adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisa yang didapat.

2.4. Klasifikasi

Klasifikasi adalah tugas dari data mining yang bertujuan untuk membangun sebuah model dari atribut kelas yang telah ditentukan sebelumnya[12]. Model yang didapat mungkin diwakili dalam berbagai format, seperti aturan klasifikasi IF-THEN, pohon keputusan, formula matematika, atau jaringan syaraf tiruan.

Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi. Aplikasi yang umum untuk klasifikasi adalah[8]:

- a. Persetujuan kredit
- b. Target marketing
- c. Diagnosa medis
- d. Analisis keefektifan tindakan

Contoh Algoritma klasifikasi yang sudah umum digunakan antara lain[8]:

- a. *Decision Tree*
- b. *Bayesian Network*
- c. *Adaptive Bayesian Network*
- d. *Naïve Bayes* dan sebagainya.

2.5. *Decision Tree*

Decision tree adalah salah satu metode klasifikasi yang paling populer karena mudah diinterpretasikan oleh manusia. *Decision tree* digunakan untuk pengenalan pola dan termasuk dalam pengenalan pola secara statistic[8]. Konsep decision tree pada dasarnya adalah mengubah data menjadi pohon keputusan dan aturan hirarki (aturan-aturan keputusan) yang pada perkembangan selanjutnya dapat disederhanakan dengan menghilangkan cabang-cabang atau aturan-aturan yang tidak perlu. Sedangkan atribut-atribut yang menyertai data tersebut menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan pohon.

2.6. *Decision Rules*

Association rules mencari hubungan antara atribut yang berbeda. *Decision rules* merupakan kasus khusus dari association rules, yang digunakan untuk klasifikasi seperti *decision tree*.

Bentuk umum dari *decision rules*: if X then Y. dengan X= *conditional attribute*, dan Y=*decision attribute*[7].

2.6.1. *Algoritma Decision Rules*

Algoritma *decision rules* yang sudah umum digunakan antara lain[7]:

- 1) Membentuk *Discernibility* Matriks Modulo D, yaitu matriks dengan $MB_{(i,j)}$ adalah sekumpulan atribut yang berbeda antara objek X_i dan X_j , dan juga berbeda *decision attribute*
- 2) Menghasilkan *reduct*
- 3) Membentuk *rule* berdasarkan hasil *reduct*

2.7. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis[9].

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja optimalisasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, *MySQL* mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, *MySQL* dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

2.7.1. Keunggulan MySQL

Ada beberapa keunggulan yang dimiliki oleh MySQL yaitu[9]:

- **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
- **Perangkat lunak sumber terbuka (*open source*).** MySQL didistribusikan sebagai *open source* sehingga dapat digunakan secara gratis.
- **Multi-user.** MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- **Performance tuning.** MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- **Ragam tipe data.** MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed* atau *unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.
- **Perintah dan Fungsi.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
- **Keamanan.** MySQL memiliki beberapa lapisan keamanan seperti *password* yang terenkripsi.
- **Skalabilitas dan Pembatasan.** MySQL mampu menangani basis data dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- **Konektivitas.** MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix socket* (UNIX), atau *named pipes* (NT).
- **Lokalisasi.** MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh

bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

- **Antar Muka.** *MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- **Klien dan Peralatan.** *MySQL* dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
- **Struktur tabel.** *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.8. Rough Set

Teori *rough set* adalah sebuah *approximation* dari suatu konsep ketidakjelasan dengan cara mencari *upper* dan *lower approximation* yang merupakan sebuah klasifikasi dari domain yang diperhatikan kedalam kategori yang dipisahkan[11]. *Table Information system* yang terdiri dari baris yang mempresentasikan data dan kolom yang mempresentasikan atribut atau variabel data. *Information system* dengan atribut tambahan yang dinamakan dengan *decision attribute*. *Lower approximation* adalah daerah yang mencakup semua elemen yang secara pasti termasuk dalam sebuah himpunan yang lain. Sedangkan *Upper approximation* adalah daerah yang mencakup semua elemen yang memiliki kemungkinan masuk dalam suatu himpunan. adalah gambaran aproksimasi himpunan yang dapat dicapai dengan menggunakan ekivalensi (atau partisi).

Diberikan ρ –Accuracy sebagai berikut[7]:

$$acc\rho = \frac{|lap_{\rho}(U)|}{|uap_{\rho}(U)|}$$

Dimana $lap_{\rho}(U) = U\{B \in \pi_{\rho} | B \subseteq U\}$

$$uap_{\rho}(U) = U\{B \in \pi_{\rho} | B \cap U \neq \emptyset\}$$

Jika $acc_{\rho}(U) = 1$, U adalah *crisp set*

$acc_{\rho}(U) < 1$, U adalah *rough set*

Misalkan x merupakan himpunan bagian dari U dan B himpunan bagian dari A maka pendekatan *lower* dan *upper* untuk B pada X didefinisikan sebagai berikut[2]:

$$\underline{B}X = \{Obj^{(i)} | Obj^{(i)} \in U, B(Obj^{(i)}) \subseteq X\}, \text{ dan}$$

$$\overline{B}X = \{Obj^{(i)} | Obj^{(i)} \in U \text{ dan } B(Obj^{(i)}) \cap X \neq \emptyset\}$$

2.8.1. Istilah Dalam *Rough Set*

Didalam *rough set* terdapat beberapa istilah yang sering digunakan diantaranya:

1. *Information System*

Tabel yang terdiri dari baris yang merepresentasikan data dan kolom yang merepresentasikan atribut atau variabel data.

2. *Decision System*

Information system dengan atribut tambahan yang dinamakan dengan *decision attribute*.

3. *Positive Region* (POS)

Matriks *Indiscernibility* dari atribut atau kombinasi atribut yang merupakan *subset indiscernibility* kelas.

$$POS_{\rho}(\rho') = \bigcup_{j=1}^n lap_{\rho}(C_j)$$

4. *Dependency Attribute* (k) *Positive*

Nilai derajat ketergantungan kombinasi variabel terhadap *positive regional* kelas.

5. *Quick Reduct*

Teknik pencarian kombinasi atribut untuk data yang jumlah variabelnya sangat besar.

2.8.2. Klasifikasi Berbasis *Rough Set*

Dalam mengklasifikasikan data historis nasabah menggunakan algoritma *rough set*, terdapat beberapa langkah diantaranya yaitu[10]:

Langkah 1: Data yang diperoleh dari Bank “X” berisi data numerik dan data kategorial. Untuk dapat melanjutkan proses algoritma berdasarkan *rough set*, semua data harus diubah ke bentuk kategorial.

Langkah 2: Menentukan *range* untuk setiap parameter agar parameter-parameter yang ada bisa terkategorialkan.

Langkah 3: Membentuk matriks *discernibility* untuk *table complete decision* diperoleh jika informasi tabel S dan *matrix discernibility* $M = (M(x,y))$ adalah sebuah matrix $|U| \times |U|$ dengan elemennya didefinisikan oleh:

$$M(x,y) = \{ a \in At / I_a(x) \neq I_a(y) \}$$

Langkah 4: Membentuk rules berdasarkan matriks *discernebility* yang diperoleh dari langkah 3.

2.8.3. Contoh Klasifikasi Berbasis *Rough Set*

Data nasabah kredit yang digunakan terdiri dari parameter-parameter, antara lain: *plafond* (P), jangka waktu (T), suku bunga (R), omzet (O), produk (S), jenis usaha (B), dan kolektibilitas (K). Misalkan, diberikan data seperti pada Gambar 2.2:

| | A | B | C | D | E | F | G | H |
|---|----|--------|-----------|------------|--------------|-----------|--|----------------|
| 1 | No | Produk | Plafond | Suku Bunga | Jangka waktu | Omzet | Jenis Usaha | Kolektibilitas |
| 2 | 1 | KI | 100000000 | 13 | 48 | 60000000 | Angkutan penumpang Dalam Kota | 1 |
| 3 | 2 | KI | 150000000 | 13 | 36 | 75000000 | Produsen Alat Olah Raga | 3 |
| 4 | 3 | KI | 150000000 | 13 | 36 | 90000000 | Perdagangan Komoditi Produk Konsumable Bukan Makanan dan Minuman | 1 |
| 5 | 4 | KI | 100000000 | 13 | 24 | 50000000 | Pedagang Bahan Konstruksi Bangunan | 2 |
| 6 | 5 | KI | 500000000 | 12 | 36 | 300000000 | Industri pengolahan kertas | 1 |

Gambar 2.2. Contoh Data Nasabah Kredit

Langkah pertama yang akan dilakukan adalah menentukan nilai *range* untuk setiap parameter. Nilai *range* ini digunakan

untuk proses konversi data ke bentuk kategorial. Untuk menentukan nilai *range* digunakan rumus:

$$Range(x) = \frac{nilai \max(x) - nilai \min(x)}{n}$$

Keterangan:

x = Nilai Parameter

$nilai \max(x)$ = Nilai maksimal pada parameter x

$nilai \min(x)$ = Nilai minimal pada parameter x

n = Jumlah kategori pada parameter x

Dengan menggunakan rumus tersebut, maka penentuan *range* tiap parameter dapat dihitung seperti pengerjaan dibawah ini.
Parameter plafond:

$$\begin{aligned} Range(P) &= \frac{nilai \max(P) - nilai \min(P)}{n} \\ &= \frac{500000000 - 100000000}{3} \\ &= 133333333 \end{aligned}$$

Sehingga didapat :

$$plafond(p) = \begin{cases} rendah; & (100000000 \leq p < 233333333) \\ standar; & (233333333 \leq p < 366666666) \\ tinggi; & (366666666 \leq p \leq 500000000) \end{cases}$$

Parameter jangka waktu:

$$\begin{aligned}
 Range(T) &= \frac{\text{nilai max}(T) - \text{nilai min}(T)}{n} \\
 &= \frac{48 - 36}{2} \\
 &= 6
 \end{aligned}$$

Sehingga didapat:

$$\text{jangka waktu}(T) = \begin{cases} \text{pendek}; & (36 \leq T < 42 \text{ bulan}) \\ \text{panjang}; & (42 \leq T \leq 48 \text{ bulan}) \end{cases}$$

Parameter suku bunga

$$\begin{aligned}
 Range(R) &= \frac{\text{nilai max}(R) - \text{nilai min}(R)}{n} \\
 &= \frac{13 - 12}{2} \\
 &= 0,5
 \end{aligned}$$

$$\text{suku bunga}(R) = \begin{cases} \text{rendah}; & (12 \leq R < 12,75 \%) \\ \text{tinggi}; & (13,75 \leq R \leq 18 \%) \end{cases}$$

Parameter omzet:

$$\begin{aligned}
 Range(O) &= \frac{\text{nilai max}(O) - \text{nilai min}(O)}{n} \\
 &= \frac{300000000 - 50000000}{3} \\
 &= 83333333
 \end{aligned}$$

Sehingga didapat:

$$omzet(O) = \begin{cases} rendah; & (50000000 \leq O < 133000000) \\ standar; & (133000000 \leq O < 216666666) \\ tinggi; & (216666666 \leq O \leq 300000000) \end{cases}$$

Setelah didapatkan nilai *range* tiap parameter-parameter diatas, maka data nasabah yang ada akan diubah ke bentuk kategorial yang dapat dilihat pada Gambar 2.3

| A1 | | fx | | No | | | |
|----|----|--------|------------|--------------|---------|---|------------------------------------|
| A | B | C | D | E | F | G | H |
| 1 | No | Produk | Suku Bunga | Jangka waktu | Omzet | Jenis Usaha | Kolektibilitas |
| 2 | 1 | KI | Rendah | Tinggi | Panjang | Rendah | Angkutan penumpang Dalam Kota |
| 3 | 2 | KI | Rendah | Tinggi | Pendek | Rendah | Produsen Alat Olah Raga |
| 3 | KI | Rendah | Tinggi | Pendek | Rendah | Perdagangan Komoditi Produk Konsumble Bukan Makanan dan Minuman | Lancar |
| 4 | 4 | KI | Rendah | Tinggi | Pendek | Rendah | Pedagang Bahan Konstruksi Bangunan |
| 5 | 5 | KI | Tinggi | Rendah | Pendek | Tinggi | Industri pengolahan kertas |
| 6 | | | | | | | Lancar |

Gambar 2.3. Contoh Data Kategorial

Langkah selanjutnya adalah membentuk matriks *discernibility* yang berukuran 5x5 karena jumlah input data yang diberikan sebanyak 5. Matriks *discernibility*nya dapat dilihat pada Gambar 2.4.

| B6 | | fx | | | | |
|----|---|-----|---------|---|---------|---------|
| | A | B | C | D | E | F |
| 1 | | 1 | 2 | 3 | 4 | 5 |
| 2 | 1 | | T,B | | T,B | |
| 3 | 2 | T,B | | B | | P,R,O,B |
| 4 | 3 | | B | | B | |
| 5 | 4 | T,B | | B | | P,R,O,B |
| 6 | 5 | | P,R,O,B | | P,R,O,B | |

Gambar 2.4. Matriks *Discernibility*

Setelah memperoleh bentuk matriks, akan dibentuk *rule* yang diperoleh berdasarkan *subset* masing-masing kolom tiap baris matriks *discernibility* sebagai berikut:

1. $(T \vee B) \wedge (T \vee B)$ sehingga $(T \vee B)$ maka $K = \text{Lancar}$
2. $(T \vee B) \wedge (B) \wedge (P \vee R \vee O \vee B)$ sehingga (B) maka $K = \text{Macet}$
3. $(B) \wedge (B)$ sehingga (B) maka $K = \text{Lancar}$
4. $(T \vee B) \wedge (B) \wedge (P \vee R \vee O \vee B)$ sehingga (B) maka $K = \text{Macet}$
5. $(P \vee R \vee O \vee B) \wedge (P \vee R \vee O \vee B)$ sehingga $(P \vee R \vee O \vee B)$ maka $K = \text{Lancar}$

Langkah terakhir membentuk *rules* yang diperoleh:

1. If $(T = \text{Panjang})$ then $(K = \text{Lancar})$
If $(B = \text{Angkutan Penumpang Dalam Kota})$ then $(K = \text{Lancar})$
2. If $(B = \text{Produsen Alat Olahraga})$ then $(K = \text{Macet})$
3. If $(B = \text{Perdagangan Komoditi Produk Konsumable Bukan Makanan dan Minuman})$ then $(K = \text{Lancar})$
4. If $(P = \text{Tinggi})$ then $(K = \text{Lancar})$
If $(R = \text{Rendah})$ then $(K = \text{Lancar})$
If $(O = \text{Tinggi})$ then $(K = \text{Lancar})$
If $(B = \text{Industri Pengolahan Kertas})$ then $(K = \text{Lancar})$
5. If $(B = \text{Persewaan Mesin Industri})$ then $(K = \text{Lancar})$

2.9. Alur Proses Permohonan Kredit Bank “X” di Surabaya

Berdasarkan data yang didapat dari bank “X”, alur proses permohonan kredit dijelaskan sebagai berikut:

1. Debitur mengajukan kredit dengan membuat surat permohonan kredit dan melengkapi persyaratan dokumen (Dokumen Legalitas Usaha, Data Jaminan, Laporan Keuangan 3 Tahun terakhir, dan Jurnal Transaksi).
2. Bagian Marketing Kredit SKC (Sentra Kredit) melakukan register dokumen dan pengecekan dokumen calon debitur.
3. Kredit Analis SKC (Sentra Kredit) melakukan analisa kredit berdasarkan data-data calon debitur dan melakukan BI *Checking*

untuk melihat historis Kolektibilitas Kreditnya. Serta melakukan kunjungan ke tempat usaha (rumah, kantor, gudang atau pabrik) untuk verifikasi obyek kredit dan jaminan.

4. Dilakukan rapat komite kredit untuk memutuskan disetujui atau ditolaknya pengajuan kredit dan mengeluarkan keputusan kredit (*Limit* yang diberikan dan keputusan bunga kredit). Jika disetujui akan diteruskan ke proses Administrasi Kredit dan jika ditolak akan dikirimkan surat penolakan kredit ke calon debitur.

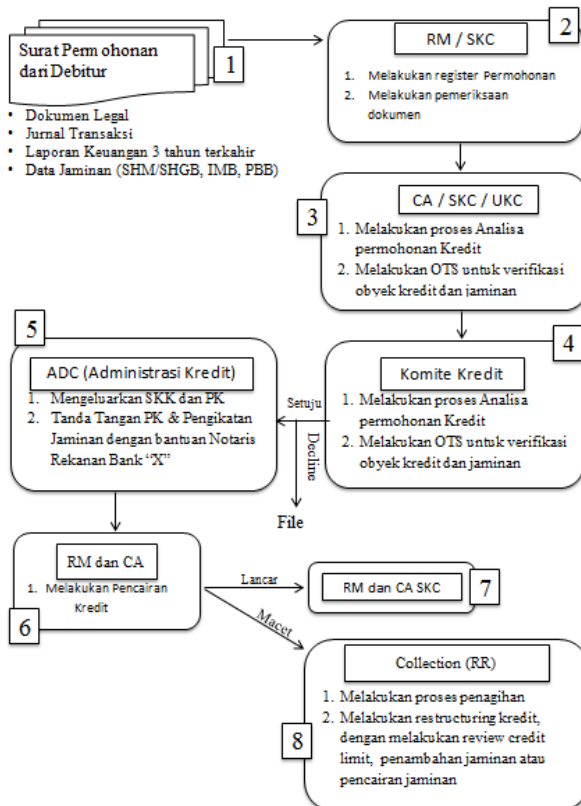
5. Proses Administrasi Kredit untuk penerbitan SKK (Surat Keputusan Kredit) dan penandatanganan PK (Perjanjian Kredit) serta dilakukan proses pengikatan jaminan kredit dengan bantuan Notaris Rekanan Bank.

6. Setelah proses administrasi kredit, dilakukan pencairan kredit ke Rekening Giro dibantu oleh bagian marketing yang akan mengawasi *performance* kredit Debitur.

7. Jika Kredit Lancar merupakan tanggung jawab RM (bagian marketing yang manage dan memantau *performance* kredit Debitur).

8. Jika Kredit Bermasalah dan Macet akan ditangani oleh bagian RR (*Remidial* dan *Recovery* Kredit) untuk dilakukan penagihan, jika debitur masih belum mampu membayar cicilan dan merasa berat dengan cicilan dan bunga akan dilakukan *Restructuring* Kredit dan *Review* Kredit. Namun jika kredit tersebut macet dan debitur tidak mampu membayar, maka akan dilakukan eksekusi jaminan kredit.

Selengkapnya ditunjukkan pada Gambar 2.5.



Gambar 2.5. Alur Proses Permohonan Kredit Bank "X"

[Hal ini sengaja dikosongkan]

BAB III

METODOLOGI PENELITIAN

Pada bab ini dijelaskan metode yang digunakan dalam tugas akhir agar proses pengerjaan dapat terstruktur dengan baik dan dapat mencapai tujuan yang telah ditetapkan sebelumnya. Maka metode penelitian yang dilakukan adalah sebagai berikut:

3.1. Metodologi Penelitian

1. Studi Pustaka

Pada langkah ini dilakukan identifikasi permasalahan, mempelajari literatur-literatur yang ada dan mengumpulkan informasi yang diperlukan meliputi *rough set*, data mining, algoritma klasifikasi *rough set* dan penilaian kredit (*credit scoring*).

2. Pengumpulan Data

Pada langkah ini akan dilakukan pengumpulan data nasabah yang diperoleh dari Bank “X” di Surabaya. Data yang dikumpulkan berupa data historis nasabah kredit Bank “X”. Parameter-parameter yang digunakan:

- Suku Bunga (R)
- *Plafond* (P)
- Kolektibilitas (K)
- Jangka waktu (T)
- Jenis Usaha (B)
- Omzet nasabah (O)
- Produk (S)

3. Pengembangan Perangkat Lunak

Pada langkah ini dilakukan *pre-processing* data, analisis dan perancangan sistem, implementasi sistem beserta evaluasi hasil kerjanya yang terdiri dari beberapa langkah berikut:

a. Pre-processing Data

Data yang didapat oleh Bank “X” berupa satu *file excel* dengan satu *sheet* atau satu tabel saja. Data ini berisi histori

nasabah kredit dengan kriteria yang dimiliki oleh masing-masing nasabah kredit beserta nilai kreditnya. Tabel terdiri dari kolom nomor, plafond, jangka waktu, suku bunga, omzet, jenis usaha, kolektibilitas, dan produk. Data mentah yang diperoleh merupakan data numerik dan data kategorial yang semuanya akan diubah menjadi data kategorial, setelah itu mengambil 90% dari data yang ada untuk dijadikan data training dan 10% dari data yang ada akan digunakan sebagai data uji. Penentuan data uji diperoleh dari *k-fold cross validation*.

b. Analisis dan Perancangan Sistem

Data kategorial yang telah terbentuk pada tahap *pre-processing* data akan disusun menjadi bentuk *table complete decision*. Bentuk *table complete decision* digunakan karena tidak adanya *missing value* dari data mentah yang diperoleh dari Bank “X”. Selanjutnya, akan dilakukan langkah-langkah algoritma klasifikasi berdasarkan *rough set* seperti yang telah dicantumkan pada Subbab 2.6.

c. Implementasi Sistem

Pada tahap ini, dilakukan implementasi algoritma serta desain dan analisis sistem yang telah dilakukan pada tahap sebelumnya. Tahap implementasi ini menggunakan bahasa pemrograman Java dengan bantuan *tool* Netbeans 8.0.2 serta beberapa *library* yang dibutuhkan.

4. Mengevaluasi Hasil Kerja dan Kesimpulan

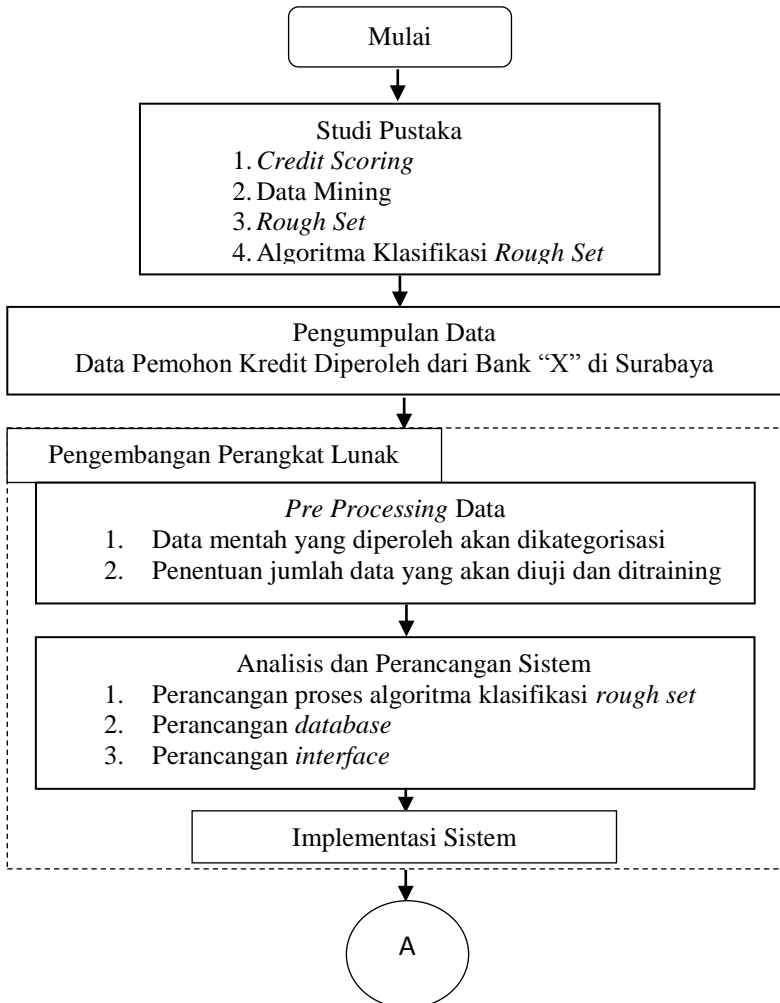
Pada langkah ini akan dilakukan evaluasi *rule* yang dihasilkan oleh algoritma klasifikasi berbasis *rough set* dengan menggunakan 1 parameter yaitu akurasi. Parameter akurasi didefinisikan sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data debitur yang terklasifikasi benar}}{\text{Jumlah data debitur keseluruhan}} \times 100\%$$

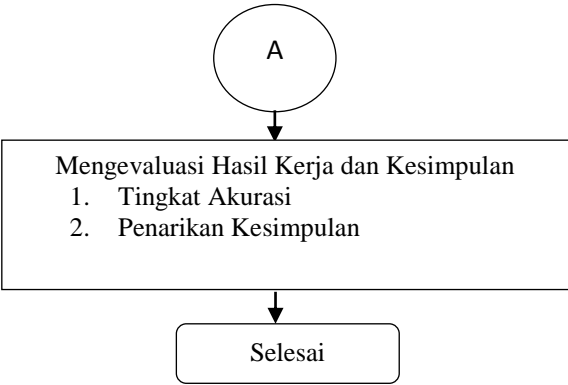
Selanjutnya akan ditarik kesimpulan merupakan tahap akhir dalam proses penelitian Tugas Akhir ini, dimana pada tahap ini dilakukan penarikan kesimpulan terhadap hasil yang telah dicapai.

3.2. Diagram Alir Penelitian

Tahap-tahap pengerjaan Tugas Akhir yang telah dijelaskan di atas digambarkan dalam diagram alir penelitian di bawah ini:



Gambar 3.1 Diagram Alir Penelitian



Gambar 3.2 Diagram Alir Penelitian (Lanjutan)

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dijelaskan mengenai pengembangan sistem yang meliputi analisis, perancangan sistem, dan implementasi proses yang telah dirancang pada perancangan sistem. Pembahasan perancangan sistem diawali dengan penjelasan analisis sistem, perancangan aplikasi *credit scoring* secara umum, perancangan data, dan perancangan antarmuka. Selanjutnya membahas implementasi sistem dan keseluruhan proses di dalam sistem.

4.1. Analisis Sistem

Aplikasi *credit scoring* merupakan program komputer yang dirancang untuk mempunyai kemampuan pemecahan masalah pemberian kredit pada calon nasabah berdasarkan data yang telah ada. Ruang lingkup masalah ini adalah penerapan algoritma klasifikasi *rough set* untuk mengetahui apakah seorang calon nasabah kredit dengan kriteria tertentu mempunyai potensi kreditnya lancar atau macet.

4.1.1. Deskripsi Perangkat Lunak

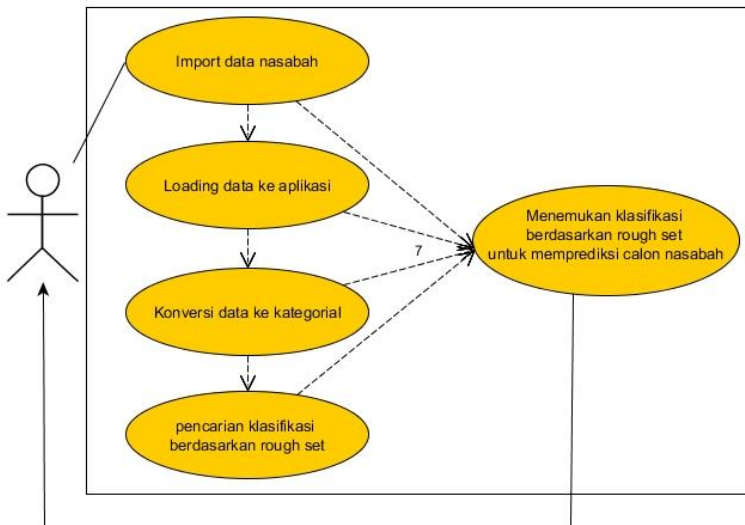
Bank seringkali mencari nasabah kredit tanpa memperhatikan jangka kedepannya apakah nasabah tersebut akan memiliki kredit macet atau lancar. Syarat-syarat yang harus dipenuhi calon nasabah bank pun seiring berjalannya waktu semakin melunak, sehingga bank membutuhkan sebuah sistem yang bisa digunakan untuk memprediksi apakah seorang calon nasabah berpotensi memiliki kredit macet atau kredit lancar. Sehingga diperlukan sebuah penyelesaian atas masalah tersebut yang bisa digunakan oleh bank untuk memprediksi calon nasabah kreditnya.

Pada penelitian Tugas Akhir ini, dilakukan pengembangan perangkat lunak yang dapat digunakan untuk mempermudah bank dalam memprediksi seorang calon nasabah kredit apakah dia

berpotensi memiliki kredit macet atau kredit lancar berdasarkan data historis yang telah dimiliki oleh bank tersebut.

4.1.2. Pemodelan Sistem

Tujuan dari perangkat lunak ini adalah mengetahui pola seorang nasabah yang kreditnya lancar dan pola seorang nasabah yang kreditnya macet berdasarkan data historis yang ada. Setelah mengetahui tingkat akurasi pola, apabila pola memiliki tingkat akurasi yang cukup tinggi pola tersebut dapat digunakan untuk memprediksi data calon nasabah. Pada Gambar 4.1 diberikan diagram *use case* perangkat lunak yang dibuat.



Gambar 4.1. Diagram *Use Case* Keseluruhan

Gambar 4.1 menjelaskan proses-proses utama pada sistem. Pada Gambar 4.1 dapat dilihat langkah pertama yang dilakukan adalah meng-*import* data nasabah, *loading* data ke aplikasi. Kemudian data nasabah dikonversi ke data kategorial, setelah itu

dilanjutkan untuk pencarian klasifikasi berbasis *rough set*. Klasifikasi berbasis *rough set*, yang terdiri dari beberapa langkah:

1. Membentuk matriks *discernibility* untuk *table complete decision* yang berdimensi $n \times n$, dimana n adalah jumlah data training.
2. Membentuk *rules* berbasis matriks *discernibility* yang diperoleh dari langkah 1.

Dari semua proses yang telah dilalui sistem, didapatkan klasifikasi berbasis *rough set* yang dapat digunakan untuk proses prediksi calon nasabah.

4.1.3. Identifikasi Permasalahan

Kredit macet merupakan suatu masalah permasalahan yang ada pada dunia perbankan. Berhasil atau tidaknya kredit seorang nasabah bisa dinilai dari kemampuan nasabah tersebut membayar tagihan kredit per bulan secara tepat waktu atau tidak. Dalam penelitian Tugas Akhir ini, akan dicari parameter yang berpotensi menjadi penentu seorang calon nasabah akan memiliki kredit macet atau lancar. Parameter yang akan diteliti diantaranya adalah:

a. *Plafond*

Plafond adalah besar kredit yang dimiliki seorang nasabah. *Plafond* pada kredit konvensional yang tergolong kredit kecil (*segmen* kredit kecil) yang limitnya max sampai 15 Milyar, oleh karena itu *plafond* akan dibagi menjadi 3 kategori yaitu *plafond* rendah, standar, dan tinggi.

b. Suku Bunga

Suku bunga adalah bunga pembayaran kredit. Jika produk nasabah kredit digolongkan kedalam kredit KI, maka suku bunga akan mengikuti. Begitu juga jika produk nasabah KMK dan TL, berlaku ketentuan yang sama.

c. Jangka Waktu

Jangka waktu adalah lama pelunasan kredit. Untuk jangka waktu mengikuti ketentuan yang sama di dalam suku bunga.

d. Omzet

Omzet adalah besar pemasukan seorang nasabah. Besar omzet untuk setiap nasabah akan berbeda satu sama lain. Omzet dikategorikan menjadi 3: rendah, standar, tinggi

e. Jenis Usaha

Jenis usaha adalah usaha yang dimiliki oleh setiap nasabah.

f. Produk

Produk adalah jenis kategori kredit yang tersedia di Bank “X” yaitu : KI, KMK, dan TL.

4.1.4. Penentuan *Range*

Sebelum dibentuk data kategorial, dibentuk terlebih dahulu *range* untuk setiap parameter. Nilai *range* akan ditunjukkan pada Tabel 4.1.

Tabel 4.1. *Range* untuk setiap Parameter

| Parameter | Nilai |
|----------------|----------------------------------|
| <i>Plafond</i> | 5.000.000 – 5.000.000.000 rupiah |
| Jangka Waktu | 12 – 120 bulan |
| Suku Bunga | 11 – 18 persen |
| Omzet | 3.000.000 – 3.000.000.000 rupiah |

4.1.5. Pembentukan Data Kategorial

Dari pembentukan *range* diatas maka parameter-parameter yang berupa data numerik bisa diubah menjadi data kategorial.

Tabel 4.2 Nilai Kategori

| Parameter | Nilai |
|----------------|-------------------------|
| <i>Plafond</i> | Rendah, Standar, Tinggi |
| Omzet | Rendah, Standar, Tinggi |
| Jangka Waktu | Pendek, Panjang |
| Suku Bunga | Rendah, Tinggi |

4.2. Perancangan Sistem

Perancangan aplikasi *credit scoring* ini menggunakan metode algoritma klasifikasi berbasis *rough set* dalam mengolah data dan mendapatkan hasil prediksi.

4.2.1. Perancangan Umum

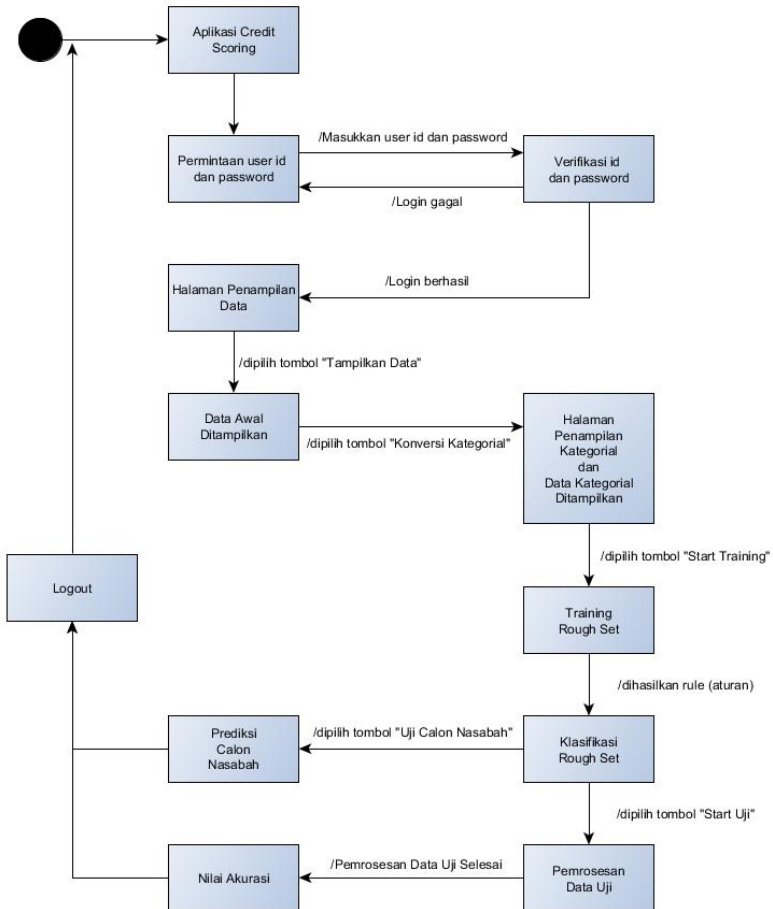
Dalam aplikasi *credit scoring* ini, rancangan umum dapat dipaparkan sebagai berikut. *Admin* yang akan menggunakan aplikasi ini memiliki *username* dan *password* yang digunakan untuk proses *login*. *Username* dan *password* sudah tersimpan dalam basisdata pengguna. *Username* dan *password* yang dimasukkan oleh pengguna pada proses *login* harus sesuai, jika tidak maka pengguna tidak akan bisa masuk ke menu utama.

Dalam halaman utama sistem, terdapat beberapa menu utama, diantaranya data nasabah, konversi data ke bentuk kategorial. Kemudian untuk menu proses *rough set*, ditampilkan proses uji data training, data uji, serta prediksi data calon nasabah, dan *logout*. Untuk data training, ada beberapa pilihan jumlah data yang akan ditraining dan sisa data nasabah yang ada untuk data uji.

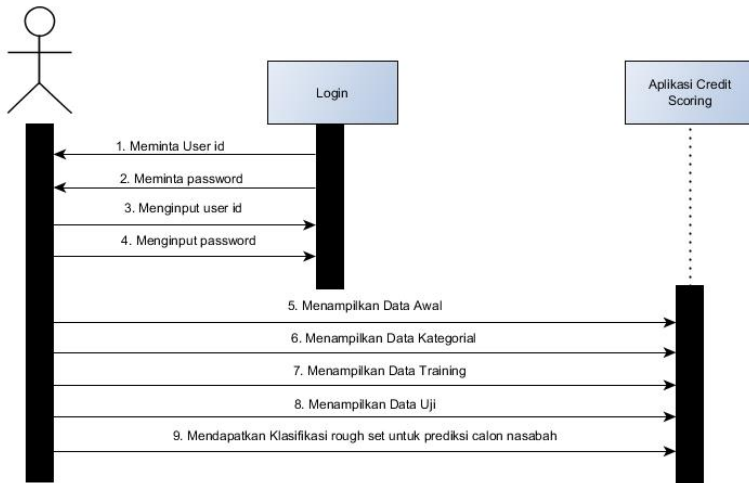
Didalam data training terdapat beberapa langkah klasifikasi berbasis *rough set*, sehingga *admin* dapat mengetahui model algoritma *rough set* dari data historis nasabah kredit yang ada untuk mengetahui karakteristik nasabah kredit yang berpotensi mengalami kemacetan kredit atau nasabah yang berpotensi lancar.

Setelah mengetahui karakteristik nasabah, *admin* dapat memprediksi calon nasabah kredit dengan memasukkan datanya pada menu prediksi calon nasabah.

Menu *logout* digunakan seorang *admin* untuk keluar dari aplikasi. *State* diagram aplikasi dan *sequence* diagram ditunjukkan oleh Gambar 4.2 dan 4.3.



Gambar 4.2. *State Diagram Aplikasi Credit Scoring*



Gambar 4.3. *Sequence Diagram Aplikasi Credit Scoring*

4.2.2. Perancangan Data

Tujuan dari perancangan data adalah untuk mendapatkan struktur data dan proses-proses yang dilakukan terhadap data tersebut. Setelah mendapatkan struktur data yang digunakan untuk menyimpan data yang diolah beserta proses-proses yang menyertainya.

Data yang didapat oleh penulis berupa satu file excel dengan satu *sheet* atau satu tabel saja. Data ini berisi data historis nasabah kredit dengan kriteria yang dimiliki oleh masing-masing nasabah beserta nilai kreditnya. Tabel terdiri dari kolom nomor, plafond, jangka waktu, suku bunga, omzet nasabah, jenis produk, jenis usaha nasabah dan kolektibilitas dengan jumlah 740 *record*. Kolektibilitas adalah nilai kredit nasabah tersebut. Apabila kolektinya 1, maka nilai kredit nasabah tersebut lancar dan apabila lebih dari 1 maka nilai kredit nasabah tersebut macet.

| | A | B | C | D | E | F | G | H |
|----|----|--------|-------------|--------------|------------|-----------|--|----------------|
| 1 | No | Produk | Plafond | Jangka Waktu | Suku Bunga | Omzet | Jenis Usaha | Kolektibilitas |
| 2 | 1 | KI | 150.000.000 | 60 | 13 | 114000000 | Rumah Makan | 1 |
| 3 | 2 | KI | 100.000.000 | 36 | 13 | 50000000 | Jasa Penyalur Asisten Rumah Tangga | 3 |
| 4 | 3 | KI | 150.000.000 | 36 | 13 | 90000000 | Pedagang Eceran Mesin dan Suku Cadang | 1 |
| 5 | 4 | KI | 100.000.000 | 36 | 13 | 60000000 | Pedagang Mainan Anak-anak | 1 |
| 6 | 5 | KI | 150.000.000 | 60 | 13 | 90000000 | Ekspedisi | 1 |
| 7 | 6 | KI | 100.000.000 | 48 | 13 | 60000000 | Angkutan penumpang dalam kota | 1 |
| 8 | 7 | KI | 150.000.000 | 36 | 13 | 75000000 | Produsen alat olahraga | 3 |
| 9 | 8 | KI | 150.000.000 | 36 | 13 | 90000000 | Perdagangan Komoditi Produk Konsumble Bukaan Makanan dan Minuman | 1 |
| 10 | 9 | KI | 500.000.000 | 36 | 12 | 300000000 | Industri Pengolahan kertas | 1 |
| 11 | 10 | KI | 175.000.000 | 36 | 13 | 105000000 | Persewaan Mesin Industri | 1 |
| 12 | 11 | KI | 300.000.000 | 36 | 13 | 180000000 | Perdagangan bahan Baku Mesin | 1 |
| 13 | 12 | KI | 150.000.000 | 36 | 13 | 90000000 | Ekspedisi | 1 |
| 14 | 13 | KI | 500.000.000 | 36 | 12 | 300000000 | Industri Peralatan Rumah Tangga | 1 |
| 15 | 14 | KI | 125.000.000 | 36 | 13 | 75000000 | Pedagangan Komoditi Makanan dan Minuman Hasil Industri | 1 |
| 16 | 15 | KI | 500.000.000 | 36 | 13 | 300000000 | SPBU | 1 |
| 17 | 16 | KI | 100.000.000 | 36 | 13 | 50000000 | Pedagang Tekstil, Pakalan Jadi dan Barang Keperluan Pribadi | 2 |
| 18 | 17 | KI | 250.000.000 | 36 | 13 | 150000000 | Produsen alat-alat musik | 1 |
| 19 | 18 | KI | 100.000.000 | 36 | 13 | 60000000 | Toko Buku, Kertas dan Peralatan Tulis Lainnya | 1 |
| 20 | 19 | KI | 150.000.000 | 36 | 13 | 60000000 | Jasa Persewaan Mobil | 1 |
| 21 | 20 | KI | 150.000.000 | 36 | 13 | 90000000 | Pedagang Eceran Mesin dan Suku Cadang | 1 |
| 22 | 21 | KI | 350.000.000 | 36 | 13 | 210000000 | Produsen alat musik | 1 |
| 23 | 22 | KI | 100.000.000 | 24 | 13 | 50000000 | Pedagang Bahan Konstruksi Bangunan | 2 |
| 24 | 23 | KI | 150.000.000 | 36 | 13 | 90000000 | Pedagangan Komoditi Makanan dan Minuman Hasil Industri | 1 |
| 25 | 24 | KI | 70.000.000 | 36 | 13 | 35000000 | Industri Kain tekstil | 2 |
| 26 | 25 | KI | 100.000.000 | 36 | 13 | 50000000 | Jasa Konsultan Arsitek dan Engineering | 2 |
| 27 | 26 | KI | 100.000.000 | 24 | 13 | 60000000 | Pedagangan Komoditi Makanan dan Minuman Hasil Industri | 1 |
| 28 | 27 | KI | 300.000.000 | 36 | 13 | 180000000 | Pedagangan Komoditi Makanan dan Minuman Hasil Industri | 1 |

Gambar 4.4. Data Nasabah Kredit pada Microsoft Excel

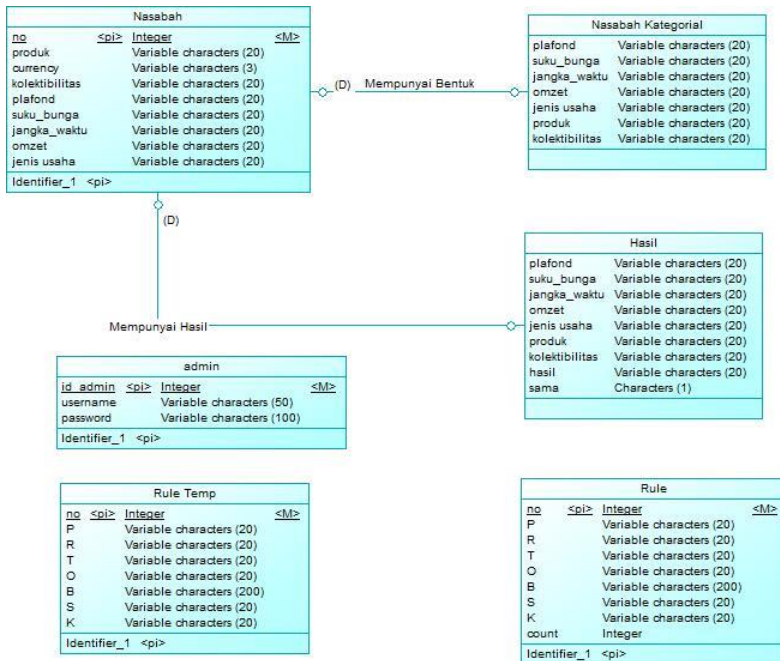
Data di atas kemudian disimpan kembali dalam format “.csv” agar bisa diimport ke DBMS MySQL. Tabel untuk data tersebut selanjutnya diberi nama **nasabah**.

4.2.2.1. Perancangan *Database*

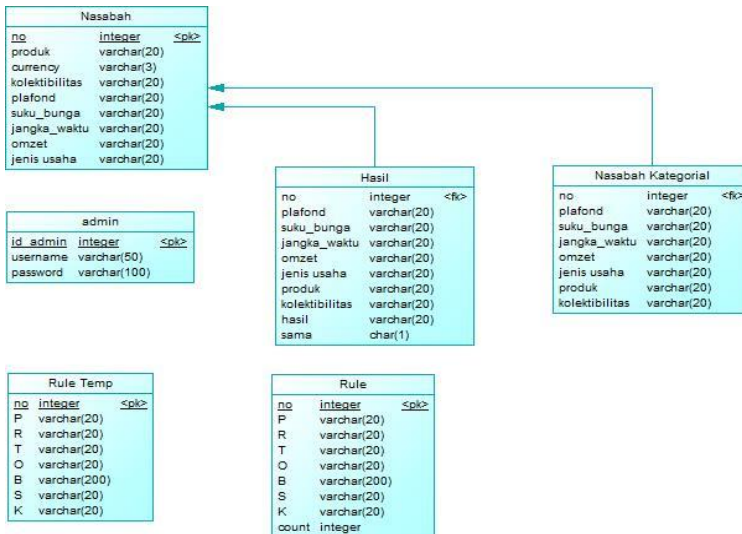
Perancangan *database* pada sistem ini meliputi *design* CDM (*Conseptual Data Model*), *design* PDM (*Physical Data Model*), dan hasil implementasi *design* pada DBMS MySQL. *Design database* menghasilkan atau bertujuan untuk mendapatkan tabel-tabel dan relasi antar tabel.

CDM (*Conseptual Data Model*) adalah tahapan untuk menganalisis struktur konsep dari suatu sistem informasi, mengidentifikasi entitas-entitas utama yang telah direpresentasikan, atribut-atribut dan relasi antar keduanya. Untuk menggambarkan CDM pada sistem ini dapat ditunjukkan pada Gambar 4.5.

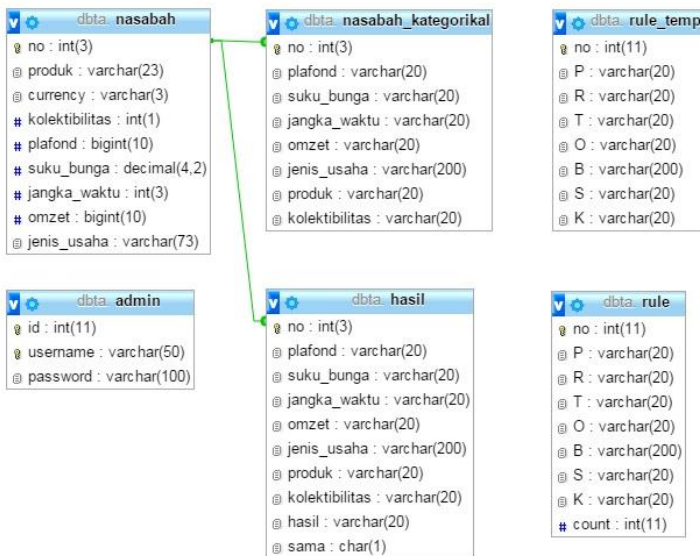
PDM (*Physical Data Model*) adalah tahapan untuk menganalisis tabel, gambaran dan objek lain dalam sebuah *database* termasuk kebutuhan objek *multidimensi* untuk data *warehouse*. Untuk menggambarkan PDM pada sistem ini dapat ditunjukkan pada Gambar 4.6.



Gambar 4.5. Design CDM Database



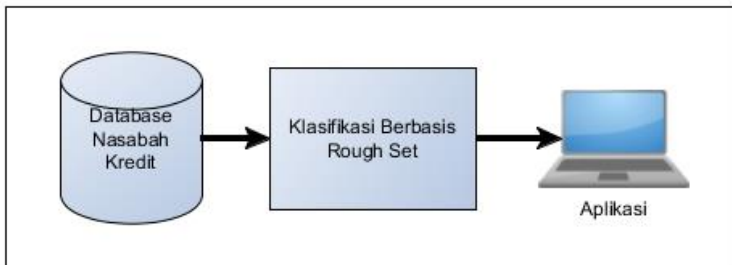
Gambar 4.6. Design PDM Database



Gambar 4.7. Implementasi Database di MySQL

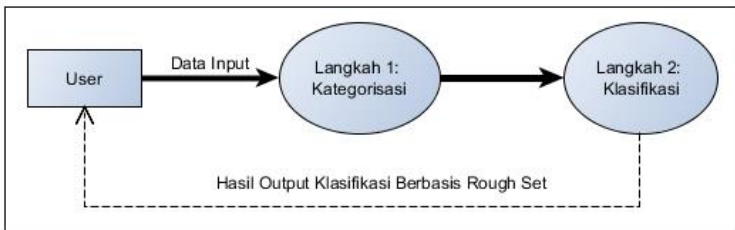
4.2.3. Perancangan Proses

Perancangan proses meliputi proses pembentukan data kategorial dari basis data awal, hingga proses pembentukan algoritma klasifikasi berbasis *rough set* yang akan digunakan dalam proses prediksi calon nasabah. Secara umum prosesnya dapat digambarkan pada Gambar 4.8.



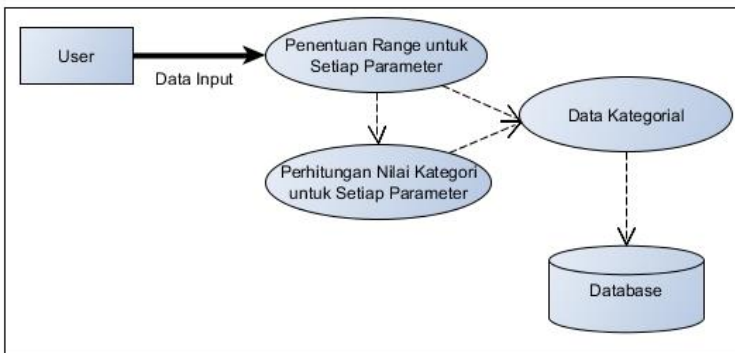
Gambar 4.8. Rancangan umum program

Dari Gambar 4.8. dapat dilihat bahwa sistem ini menggunakan database sebagai media penyimpanan data. Untuk kebutuhan pengolahan data yang lebih terpusat pada tujuan Tugas Akhir ini, data asli nasabah kredit yang diperoleh dari Bank “X” dilakukan penyaringan atau filter parameter-parameter yang diperlukan saja. Untuk proses klasifikasi *rough set* dapat dilihat pada Gambar 4.9.

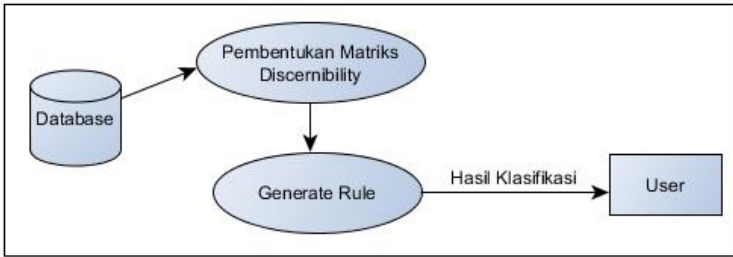


Gambar 4.9. Proses klasifikasi berbasis *rough set*

Dari Gambar 4.9. dapat dilihat proses klasifikasi berbasis *rough set*. Langkah pertama yaitu kategorisasi, data mentah yang diperoleh dari Bank akan diubah menjadi bentuk kategorial. Proses pengubahan data awal ke dalam bentuk kategorial dilakukan dengan cara seperti yang ada pada Gambar 4.10. Langkah-langkah proses klasifikasi dapat dilihat pada Gambar 4.11.



Gambar 4.10. Proses pembentukan data kategorial



Gambar 4.11. Langkah-langkah proses klasifikasi

Untuk mengubah data nasabah ke dalam data kategorial diperlukan penentuan nilai *range* untuk setiap parameter yang ada. Nilai *range* ini digunakan untuk proses konversi data ke bentuk kategorial. Untuk menentukan nilai *range* digunakan rumus:

$$Range(x) = \frac{nilai\ max(x) - nilai\ min(x)}{n}$$

Keterangan:

x = Nilai Parameter

$nilai\ max(x)$ = Nilai maksimal pada parameter x

$nilai\ min(x)$ = Nilai minimal pada parameter x

n = Jumlah kategori pada parameter x

Dengan menggunakan rumus tersebut, maka penentuan *range* tiap parameter dapat dihitung seperti pengerjaan dibawah ini.
Parameter plafond:

$$\begin{aligned}
 Range(P) &= \frac{nilai\ max(P) - nilai\ min(P)}{n} \\
 &= \frac{5000000000 - 5000000}{3} \\
 &= 1665000000
 \end{aligned}$$

Sehingga didapat :

$$plafond(p) = \begin{cases} \text{rendah}; & (5000000 \leq p < 1670000000) \\ \text{standar}; & (1670000000 \leq p < 3335000000) \\ \text{tinggi}; & (3335000000 \leq p \leq 5000000000) \end{cases}$$

Parameter jangka waktu:

$$\begin{aligned} Range(T) &= \frac{\text{nilai max}(T) - \text{nilai min}(T)}{n} \\ &= \frac{120 - 12}{2} \\ &= 54 \end{aligned}$$

Sehingga didapat:

$$jangka\ waktu(T) = \begin{cases} \text{pendek}; & (12 \leq T < 66 \text{ bulan}) \\ \text{panjang}; & (66 \leq T \leq 120 \text{ bulan}) \end{cases}$$

Parameter suku bunga

$$\begin{aligned} Range(R) &= \frac{\text{nilai max}(R) - \text{nilai min}(R)}{n} \\ &= \frac{18 - 11}{2} \\ &= 3,5 \end{aligned}$$

$$suku\ bunga(R) = \begin{cases} \text{rendah}; & (12 \leq R < 15,5 \%) \\ \text{tinggi}; & (15,5 \leq R \leq 18 \%) \end{cases}$$

Parameter omzet:

$$Range(O) = \frac{\text{nilai max}(O) - \text{nilai min}(O)}{n}$$

$$\begin{aligned}
&= \frac{3000000000 - 3000000}{3} \\
&= 999000000
\end{aligned}$$

Sehingga didapat:

$$omzet(O) = \begin{cases} rendah; & (3000000 \leq O < 1002000000) \\ standar; & (1002000000 \leq O < 2001000000) \\ tinggi; & (2001000000 \leq O \leq 3000000000) \end{cases}$$

Setelah mendapatkan nilai *range* tiap parameter-parameter yang ada, maka selanjutnya semua data mentah yang diperoleh bisa dikategorisasi. Langkah kedua dalam proses klasifikasi berbasis *rough set* yaitu dengan membentuk matriks *discernibility*.

Untuk membentuk matriks *discernibility* yaitu dengan menginisialisasi matriks berukuran $n \times n$, n adalah jumlah data training. Setelah itu untuk setiap data training akan dilakukan:

1. Data ke- i dicocokkan dengan data ke- $(i+1)$;
2. Jika nilai kolektibilitas (K) berbeda, maka:
Pilih parameter-parameter yang berbeda nilai dan simpan ke matriks $[i][i+1]$ dan matriks $[i+1][i]$;
Jika nilai kolektibilitas (K) sama, maka:
Lanjutkan ke data selanjutnya;
3. Dilanjutkan dengan proses pembentukan *rule*;

Proses pembentukan *rule* untuk setiap baris ke- i pada matriks:

1. Untuk setiap baris ke- i pada matriks
 - Setiap elemen pada baris ke- i dioperasikan sehingga menghasilkan himpunan parameter yang merupakan subset dari keseluruhan elemen pada baris tersebut;
 - Himpunan parameter yang dihasilkan kemudian dikombinasikan untuk dijadikan *rule* yang mempunyai nilai

kolektibilitas sama dengan nilai kolektibilitas pada data training ke-i;

Dari proses matriks *discernibility* yang telah dibentuk, menghasilkan beberapa *rules*, antara lain:

1. If (T=Panjang) then (K=Lancar)
If (B=Angkutan Penumpang Dalam Kota) then (K=Lancar)
2. If (B=Produsen Alat Olahraga) then (K=Macet)
3. If (P=Standar) then (K=Lancar)
If (R=Rendah) then (K=Lancar)
If (B=Penyalur Kredit Non Leasing) then (K=Lancar)
4. If (P=Tinggi) then (K=Lancar)
If (R=Rendah) then (K=Lancar)
If (O=Tinggi) then (K=Lancar)
If (B=Industri Pengolahan Kertas) then (K=Lancar)
5. If (B=Persewaan Mesin Industri) then (K=Lancar)
6. If (P=Standar) then (K=Lancar)
If (R=Rendah) then (K=Lancar)
If (O=Standar) then (K=Lancar)
7. If (O=Rendah) then (K=Lancar)
If (B=Event Organizer) then (K=Lancar)
- :
- :
- :
661. If (P=Tinggi) then (K=Lancar)
If (R=Tinggi) then (K=Lancar)
If (T=Panjang) then (K=Lancar)
If (O=Rendah) then (K=Lancar)

Langkah 3 pada proses klasifikasi berbasis *rough set* yaitu uji data. Dengan *rules* yang telah dihasilkan, maka kita dapat menggunakannya untuk menguji data calon nasabah kredit apakah layak menerima kredit atau tidak.

4.3. Implementasi Sistem

Berdasarkan perancangan sistem yang telah dipaparkan maka pada subbab ini akan dijelaskan proses implementasinya.

4.3.1. Lingkungan Implementasi

Lingkungan implementasi ini merupakan pengimplementasian algoritma klasifikasi berbasis *rough set* dalam prediksi calon nasabah kredit di Bank “X” di Surabaya kedalam bahasa pemrograman. Sistem di bangun pada lingkungan implementasi perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut merupakan lingkungan implementasi perangkat keras dan perangkat lunak yang digunakan.

4.3.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam implementasi aplikasi *Rough Set* adalah

1. Prosesor Intel(R) Core(TM) i5-3337U CPU @1.80 GHz
2. Memori 4 GB RAM
3. Hard Disk 750 GB

4.3.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk implementasi *Rough Set* adalah:

1. Sistem Operasi Windows 8 Ultimate 64-bit
2. NetBeans IDE 8.0.2 untuk pembangunan sistem
3. *Open source* MySQL 5.6 sebagai sistem manajemen basis datanya.

4.3.2. Implementasi Program

Berdasarkan perancangan sistem yang telah dipaparkan maka pada subbab ini akan dijelaskan proses implementasinya.

4.3.2.1. Implementasi Konversi Kategorial

Proses pembentukan matriks *discernibility* dimulai dengan mengambil data nasabah dari Bank ”X” dengan membaca

file *excel* yang berformat *.xls*. setelah melakukan baca file excel, kemudian meng-*import* data nasabah dan *loading* data ke aplikasi. Kemudian data nasabah dikonversi ke data kategorial. Proses konversi kategorial dapat dilihat pada Lampiran *Source Code* Konversi Kategorial.

4.3.2.2. Implementasi Matriks *Discernibility*

Tahap setelah konversi ke data kategorial, dilanjutkan untuk pembentukan matriks *discernibility*. Proses matriks *discernibility* dapat dilihat pada Lampiran *Source Code* Matriks *Discernibility*.

4.3.2.3. Implementasi *Rules*

Matriks *discernibility* yang dibentuk menghasilkan *rules-rules*. Untuk proses implementasi *rules-rules* yang telah dihasilkan dapat dilihat pada *source code* berikut.

```
String[] dataRule=data.split("&");
String query1="insert into rule_temp(";
String query2="";
for(int i=0;i<dataRule.length;i++){
    String nilai[]=dataRule[i].split("=");
    query2+=nilai[0];
    if(i<dataRule.length-1){
        query2+=",";
    }
}
query2+=") values(";
for(int i=0;i<dataRule.length;i++){
    String nilai[]=dataRule[i].split("=");
    query2=query2+"'" +nilai[1].trim()+"'";
    if(i<dataRule.length-1){
        query2+=",";
    }
}
```

```

        query2+=")";
        String query=query1+query2;
        System.out.println(query);

        try {
            stm.executeUpdate(query);
        } catch (SQLException ex) {

            Logger.getLogger(FKategorikal.class.getName()).log(
            Level.SEVERE, null, ex);
        }
    }
    String query="INSERT INTO rule(P, R, T, O, B, S, K,
count) SELECT P, R, T, O, B, S, K, COUNT(1)
FROM rule_temp GROUP BY P, R, T,O,B,S,K";
        stm.executeUpdate(query);
        System.out.println("data sudah dimasukkan ke
rule");
    } catch (SQLException ex) {

        Logger.getLogger(FKategorikal.class.getName()).log(
        Level.SEVERE, null, ex);
    }
}

```

4.3.2.4. Implementasi *Rough Set* Uji Data

Tahap terakhir setelah mendapatkan *rules-rules* yaitu dengan dilakukan uji data. Proses uji data dapat dilihat pada Lampiran *Source Code* Uji Data.

[Hal ini sengaja dikosongkan]

BAB V

UJI COBA DAN PEMBAHASAN

Bab ini membahas mengenai hasil dan pengujian dari prediksi calon nasabah kredit dengan menggunakan algoritma klasifikasi berbasis *rough* set yang telah dibuat. Pembahasan pengujian ini meliputi ruang lingkup uji coba.

5.1. Ruang Lingkup Uji Coba

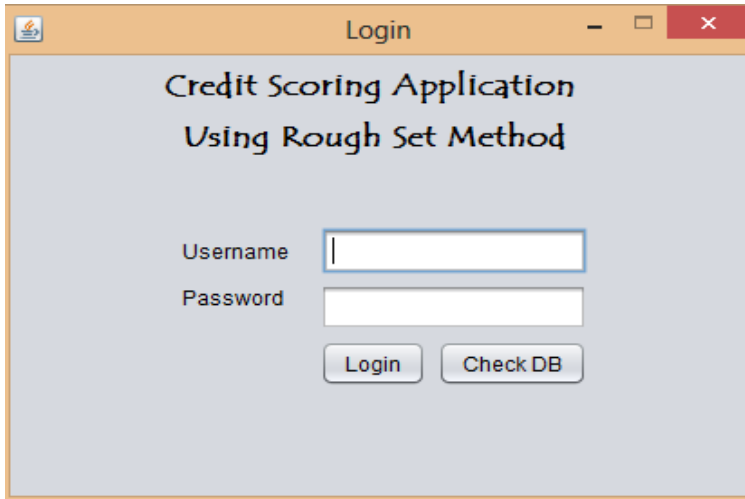
Ruang lingkup implementasi program yang dibangun dalam Tugas Akhir ini meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasinya disajikan pada Tabel 5.1.

Tabel 5.1. Ruang Lingkup Uji Coba

| | |
|-----------------|---|
| Perangkat Keras | <ol style="list-style-type: none">1. Prosesor Intel(R) Core(TM) i5-3337U CPU @1.80 GHz 1.80 GHz2. Memori 4 GB RAM3. Hard Disk 750 Gb |
| Perangkat Lunak | <ol style="list-style-type: none">1. Sistem Operasi Windows 8 Ultimate 64-bit2. NetBeans IDE 8.0.2 untuk pembangunan sistem3. <i>open source</i> MySQL 5.6 sebagai sistem manajemen basis datanya |

5.2. Uji Tampilan Awal

Halaman ini merupakan tampilan awal dari aplikasi penilaian kredit.

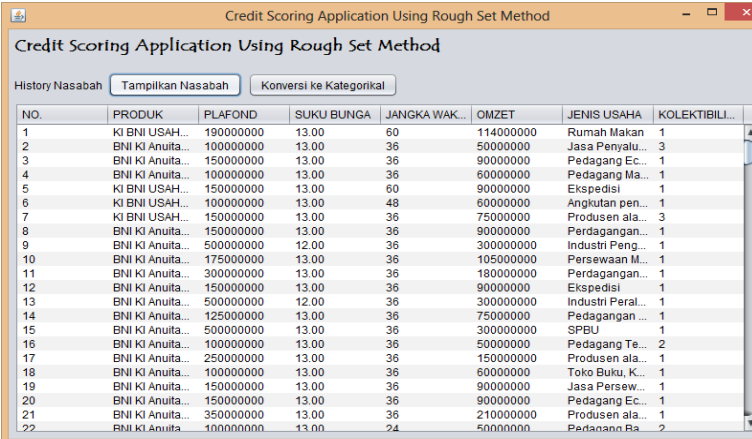
The image shows a Windows-style window titled "Login". Inside the window, the text "Credit Scoring Application" and "Using Rough Set Method" is displayed in a serif font. Below this, there are two input fields: "Username" and "Password". The "Username" field has a blue border and a cursor. Below the input fields are two buttons: "Login" and "Check DB". The window has a standard Windows title bar with minimize, maximize, and close buttons.

Gambar 5.1. Halaman *Login*

Pada halaman ini user melakukan *login* dengan cara memasukkan *username* beserta *password* dan kemudian memilih tombol “Login”

5.3. Uji halaman penampilan data

Halaman ini tampil setelah user berhasil *login*



Credit Scoring Application Using Rough Set Method

History Nasabah Tampilkan Nasabah Konversi ke Kategorikal

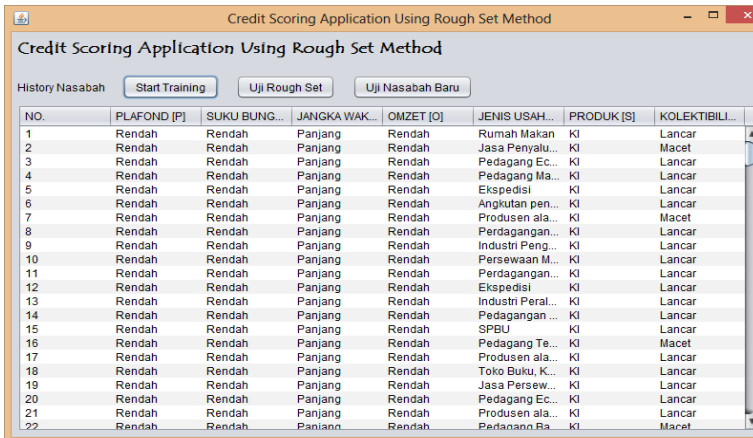
| NO. | PRODUK | PLAFOND | SUKU BUNGA | JANGKA WAK... | OMZET | JENIS USAHA | KOLEKTIBILIT... |
|-----|------------------|-----------|------------|---------------|-----------|-------------------|-----------------|
| 1 | KI BNI USAH... | 190000000 | 13.00 | 60 | 114000000 | Rumah Makan | 1 |
| 2 | BNI KI Anuita... | 100000000 | 13.00 | 36 | 50000000 | Jasa Penyalu... | 3 |
| 3 | BNI KI Anuita... | 150000000 | 13.00 | 36 | 90000000 | Pedagang Ec... | 1 |
| 4 | BNI KI Anuita... | 100000000 | 13.00 | 36 | 60000000 | Pedagang Ma... | 1 |
| 5 | KI BNI USAH... | 150000000 | 13.00 | 60 | 90000000 | Ekspedisi | 1 |
| 6 | KI BNI USAH... | 100000000 | 13.00 | 48 | 60000000 | Angkutan pen... | 1 |
| 7 | KI BNI USAH... | 150000000 | 13.00 | 36 | 75000000 | Produsen ala... | 3 |
| 8 | BNI KI Anuita... | 150000000 | 13.00 | 36 | 90000000 | Perdagangan... | 1 |
| 9 | BNI KI Anuita... | 500000000 | 12.00 | 36 | 300000000 | Industri Peng... | 1 |
| 10 | BNI KI Anuita... | 175000000 | 13.00 | 36 | 105000000 | Persewa M... | 1 |
| 11 | BNI KI Anuita... | 300000000 | 13.00 | 36 | 180000000 | Perdagangan... | 1 |
| 12 | BNI KI Anuita... | 150000000 | 13.00 | 36 | 90000000 | Ekspedisi | 1 |
| 13 | BNI KI Anuita... | 500000000 | 12.00 | 36 | 300000000 | Industri Peral... | 1 |
| 14 | BNI KI Anuita... | 125000000 | 13.00 | 36 | 75000000 | Pedagangan ... | 1 |
| 15 | BNI KI Anuita... | 500000000 | 13.00 | 36 | 300000000 | SPBU | 1 |
| 16 | BNI KI Anuita... | 100000000 | 13.00 | 36 | 50000000 | Pedagang Te... | 2 |
| 17 | BNI KI Anuita... | 250000000 | 13.00 | 36 | 150000000 | Produsen ala... | 1 |
| 18 | BNI KI Anuita... | 100000000 | 13.00 | 36 | 60000000 | Toko Buku, K... | 1 |
| 19 | BNI KI Anuita... | 150000000 | 13.00 | 36 | 90000000 | Jasa Persew... | 1 |
| 20 | BNI KI Anuita... | 150000000 | 13.00 | 36 | 90000000 | Pedagang Ec... | 1 |
| 21 | BNI KI Anuita... | 350000000 | 13.00 | 36 | 210000000 | Produsen ala... | 1 |
| 22 | BNI KI Anuita... | 100000000 | 13.00 | 24 | 50000000 | Perdagangan Ba... | 2 |

Gambar 5.2. Halaman Penampilan Data

Pada halaman ini ditampilkan data yang akan melalui proses mining. Untuk mengubah data ke dalam bentuk kategorial, *admin* dapat memilih tombol “Konversi ke Kategorikal”

5.4. Uji halaman data yang telah diubah kedalam bentuk kategorial

Pada halaman ini ditampilkan data yang sudah dalam bentuk kategorial.



Credit Scoring Application Using Rough Set Method

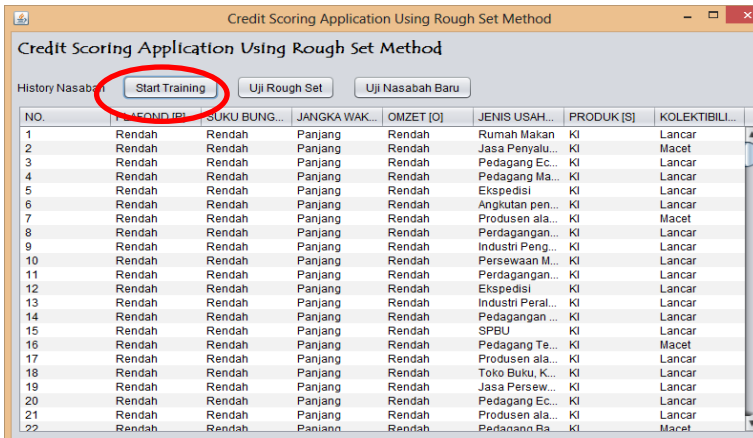
History Nasabah **Start Training** Uji Rough Set Uji Nasabah Baru

| NO. | PLAFOND [P] | SUKU BUNG... | JANGKA WAK... | OMZET [O] | JENIS USAH... | PRODUK [S] | KOLEKTIBILI... |
|-----|-------------|--------------|---------------|-----------|-------------------|------------|----------------|
| 1 | Rendah | Rendah | Panjang | Rendah | Rumah Makan | KI | Lancar |
| 2 | Rendah | Rendah | Panjang | Rendah | Jasa Penyalu... | KI | Macet |
| 3 | Rendah | Rendah | Panjang | Rendah | Pedagang Ec... | KI | Lancar |
| 4 | Rendah | Rendah | Panjang | Rendah | Pedagang Ma... | KI | Lancar |
| 5 | Rendah | Rendah | Panjang | Rendah | Ekspedisi | KI | Lancar |
| 6 | Rendah | Rendah | Panjang | Rendah | Angkutan pen... | KI | Lancar |
| 7 | Rendah | Rendah | Panjang | Rendah | Produsen ala... | KI | Macet |
| 8 | Rendah | Rendah | Panjang | Rendah | Perdagangan... | KI | Lancar |
| 9 | Rendah | Rendah | Panjang | Rendah | Industri Peng... | KI | Lancar |
| 10 | Rendah | Rendah | Panjang | Rendah | Persewaan M... | KI | Lancar |
| 11 | Rendah | Rendah | Panjang | Rendah | Perdagangan... | KI | Lancar |
| 12 | Rendah | Rendah | Panjang | Rendah | Ekspedisi | KI | Lancar |
| 13 | Rendah | Rendah | Panjang | Rendah | Industri Peral... | KI | Lancar |
| 14 | Rendah | Rendah | Panjang | Rendah | Pedagangan ... | KI | Lancar |
| 15 | Rendah | Rendah | Panjang | Rendah | SPBU | KI | Lancar |
| 16 | Rendah | Rendah | Panjang | Rendah | Pedagang Te... | KI | Macet |
| 17 | Rendah | Rendah | Panjang | Rendah | Produsen ala... | KI | Lancar |
| 18 | Rendah | Rendah | Panjang | Rendah | Toko Buku, K... | KI | Lancar |
| 19 | Rendah | Rendah | Panjang | Rendah | Jasa Persew... | KI | Lancar |
| 20 | Rendah | Rendah | Panjang | Rendah | Pedagang Ec... | KI | Lancar |
| 21 | Rendah | Rendah | Panjang | Rendah | Produsen ala... | KI | Lancar |
| 22 | Rendah | Rendah | Panjang | Rendah | Perdagangan Ba... | KI | Macet |

Gambar 5.3. Halaman Penampilan Data Kategorial

5.5. Uji Halaman Data Training

Pada halaman penampilan data kategorial, tekan tombol Start Training untuk memulai proses klasifikasi *Rough Set* yang menghasilkan *rule*.



Credit Scoring Application Using Rough Set Method

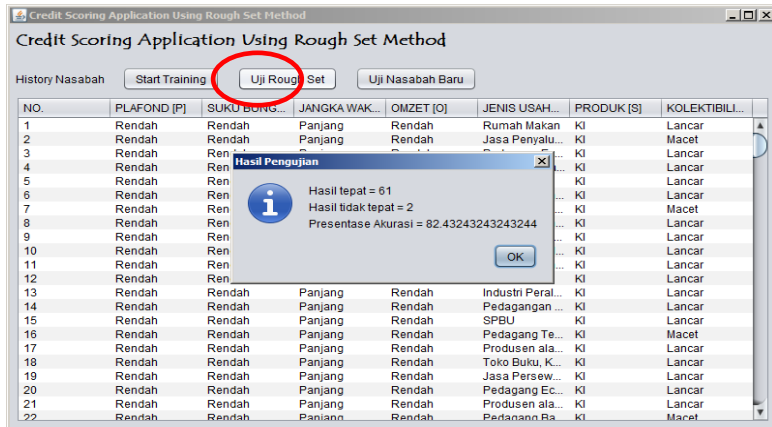
History Nasabah **Start Training** Uji Rough Set Uji Nasabah Baru

| NO. | PLAFOND [P] | SUKU BUNG... | JANGKA WAK... | OMZET [O] | JENIS USAH... | PRODUK [S] | KOLEKTIBILI... |
|-----|-------------|--------------|---------------|-----------|-------------------|------------|----------------|
| 1 | Rendah | Rendah | Panjang | Rendah | Rumah Makan | KI | Lancar |
| 2 | Rendah | Rendah | Panjang | Rendah | Jasa Penyalu... | KI | Macet |
| 3 | Rendah | Rendah | Panjang | Rendah | Pedagang Ec... | KI | Lancar |
| 4 | Rendah | Rendah | Panjang | Rendah | Pedagang Ma... | KI | Lancar |
| 5 | Rendah | Rendah | Panjang | Rendah | Ekspedisi | KI | Lancar |
| 6 | Rendah | Rendah | Panjang | Rendah | Angkutan pen... | KI | Lancar |
| 7 | Rendah | Rendah | Panjang | Rendah | Produsen ala... | KI | Macet |
| 8 | Rendah | Rendah | Panjang | Rendah | Perdagangan... | KI | Lancar |
| 9 | Rendah | Rendah | Panjang | Rendah | Industri Peng... | KI | Lancar |
| 10 | Rendah | Rendah | Panjang | Rendah | Persewaan M... | KI | Lancar |
| 11 | Rendah | Rendah | Panjang | Rendah | Perdagangan... | KI | Lancar |
| 12 | Rendah | Rendah | Panjang | Rendah | Ekspedisi | KI | Lancar |
| 13 | Rendah | Rendah | Panjang | Rendah | Industri Peral... | KI | Lancar |
| 14 | Rendah | Rendah | Panjang | Rendah | Pedagangan ... | KI | Lancar |
| 15 | Rendah | Rendah | Panjang | Rendah | SPBU | KI | Lancar |
| 16 | Rendah | Rendah | Panjang | Rendah | Pedagang Te... | KI | Macet |
| 17 | Rendah | Rendah | Panjang | Rendah | Produsen ala... | KI | Lancar |
| 18 | Rendah | Rendah | Panjang | Rendah | Toko Buku, K... | KI | Lancar |
| 19 | Rendah | Rendah | Panjang | Rendah | Jasa Persew... | KI | Lancar |
| 20 | Rendah | Rendah | Panjang | Rendah | Pedagang Ec... | KI | Lancar |
| 21 | Rendah | Rendah | Panjang | Rendah | Produsen ala... | KI | Lancar |
| 22 | Rendah | Rendah | Panjang | Rendah | Perdagangan Ba... | KI | Macet |

Gambar 5.4. Halaman Penampilan Uji Data Training

5.6. Halaman Data Uji

Pada halaman penampilan data kategorial, tekan tombol Uji *Rough Set* untuk memulai proses data uji *rough set* yang akan menampilkan nilai akurasi.



Gambar 5.5. Halaman Penampilan Uji

5.7. Halaman Uji Calon Nasabah Kredit

Pada halaman ini ditampilkan form yang berisi *plafond*, jangka waktu, suku bunga, omzet, produk, dan lama usaha. Halaman ini berfungsi untuk memprediksi apakah seorang calon nasabah berpotensi memiliki kredit lancar atau macet. Cara untuk memprediksi calon nasabah tersebut adalah dengan memasukkan nilai pada masing-masing tempat yang telah disediakan. Setelah semua parameter sudah diisi, user memilih tombol “Uji Rough Set” untuk menampilkan hasil prediksi.

Misalkan ada calon nasabah dengan pengajuan plafond sebesar 350000000, jangka waktu 36 bulan, suku bunga 12%, omzet 250000000, produk KI, dan jenis usaha rumah sakit, didapat hasil prediksi macet.

Credit Scoring Application Using Rough Set Method

Silakan masukkan rencana pengajuan calon nasabah

| | | | |
|--------------|--|--------|--|
| Plafond | <input type="text" value="350000000"/> | Omzet | <input type="text" value="200000000"/> |
| Suku_Bunga | <input type="text" value="12"/> | Produk | <input type="text" value="KI"/> |
| Jangka_Waktu | <input type="text" value="36"/> | | |

Jenis_Usaha

Uji Rough Set

Macet

Gambar 5.6. Halaman Contoh Prediksi Calon Nasabah Macet

Dan apabila seorang calon nasabah dengan pengajuan plafond sebesar 800000000, jangka waktu 120 bulan, suku bunga 12%, omzet 400000000, produk KI, dan jenis usaha rumah makan, didapat hasil prediksi lancar.

Credit Scoring Application Using Rough Set Method

Silakan masukkan rencana pengajuan calon nasabah

Plafond: 80000000 Omzet: 40000000

Suku_Bunga: 12 Produk: KI

Jangka_Waktu: 120

Jenis_Usaha: Rumah Makan

Uji Rough Set

Lancar

Gambar 5.7. Halaman Contoh Prediksi Calon Nasabah Lancar

5.8. Uji Coba

Dalam uji coba algoritma klasifikasi berbasis *rough set* ini, terdapat 2 pengujian dilakukan yaitu:

1. Uji coba dengan *k-fold cross validation*
2. Uji coba 90% dari data nasabah akan menjadi data training, dan 10% data akan menjadi data uji.

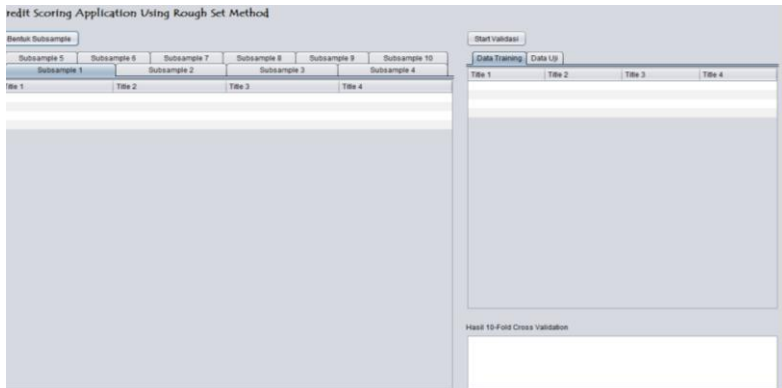
Uji coba yang pertama dengan *k-fold cross validation* dengan $k=10$, maka terdapat 10 kali uji coba. Untuk proses *k-fold validation* dapat dilihat pada Tabel berikut.

Tabel 5.2. Data Uji *K-Fold Cross Validation*

| | | |
|--------|-------|---------|
| Data 1 | | Data 10 |
|--------|-------|---------|

Data ke-1 akan menjadi data uji, dan sisanya sampai data ke-10 menjadi data training. Proses *k-fold cross validation* masih

terus dilakukan sampai data 10 menjadi data uji, dan sisanya menjadi data training. Proses *k-fold cross validation* dalam sistem dapat dilihat pada Gambar 5.8



Gambar 5.8. Halaman Uji *K-Fold Cross Validation*

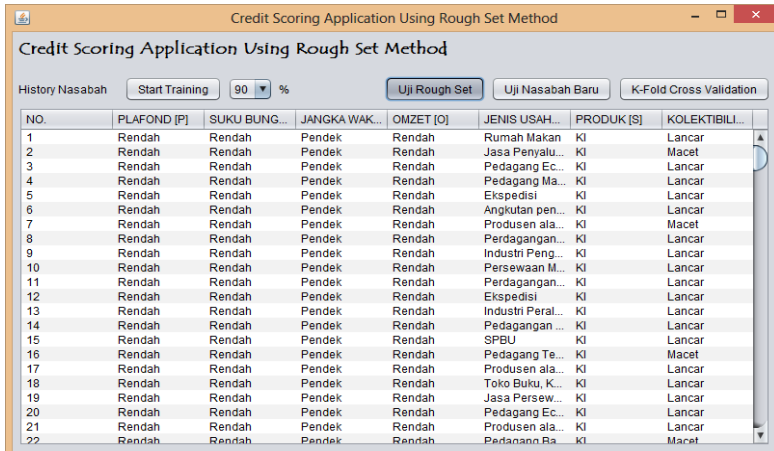
Setelah menguji ke 10 data dengan menggunakan *k-fold cross validation*, didapatkan nilai akurasi tiap-tiap data yang dapat dilihat pada Tabel berikut.

Tabel 5.3. Nilai akurasi *k-fold cross validation*

| Uji Ke | Nilai Akurasi |
|--------|-------------------|
| 1 | 78.37837837837837 |
| 2 | 82.43243243243244 |
| 3 | 86.48648648648648 |
| 4 | 86.48648648648648 |
| 5 | 86.48648648648648 |
| 6 | 81.08108108108108 |
| 7 | 86.48648648648648 |
| 8 | 83.78378378378379 |
| 9 | 77.02702702702703 |
| 10 | 83.78378378378379 |

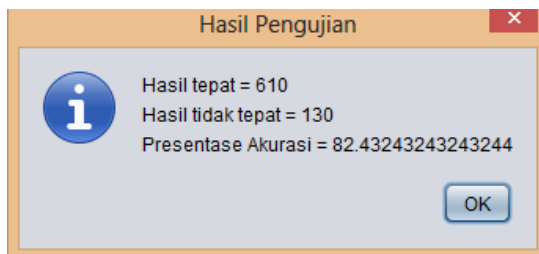
Rata-rata yang didapatkan dengan nilai akurasi-akurasi yang ada : 82.3%.

Uji coba yang kedua dengan 90% data training dan 10% data uji. Proses uji coba dalam sistem dapat dilihat pada Gambar 5.9.



Gambar 5.9. Halaman Proses Uji coba

Nilai akurasi yang didapat untuk uji coba yang kedua yaitu sebesar 82,4%. Dapat dilihat pada Gambar 5.10



Gambar 5.10. Halaman Nilai Akurasi

[hal ini sengaja dikosongkan]

BAB VI PENUTUP

Pada bab ini dipaparkan kesimpulan yang dapat diambil dari Tugas Akhir ini beserta saran penulis jika penelitian ini dimaksudkan untuk dikembangkan lebih lanjut.

6.1. Kesimpulan

Berdasarkan analisis terhadap hasil pengujian sistem aplikasi penilai kredit pada Bank “X” menggunakan algoritma *rough set*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem ini telah berhasil menampilkan data yang diproses dan diubah ke dalam bentuk kategorial dengan proses klasifikasi berdasarkan *rough set* untuk masing-masing parameter.
2. Algoritma *rough set* mampu menghasilkan aturan yang memiliki tingkat akurasi 82,4% untuk 90% data training dan 10% data uji.

6.1. Saran

Dengan melihat hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya, yaitu untuk mendapatkan hasil yang lebih baik, perlu dilakukan uji coba dengan sistem yang lebih efektif seperti mengkombinasikan metode *Rough Set* dengan metode lain yang dapat saling mendukung, contohnya *Fuzzy Rough Set*.

[halaman ini sengaja dikosongkan]

LAMPIRAN

➤ *Source Code konversi Kategorial*

```
private void
btnKonversiKategorikalActionPerformed(java.awt.event.ActionEvent
evt) {
    ConnectDB connect = new ConnectDB();
    Statement stm = null;
    Connection con = null;
    connect.Connect();
    con = connect.getConnection();
    stm = connect.createStatement();
    int jumBaris = tblNasabah.getRowCount();
    String hproduk = null, hplafond = null, homzet = null, hkolek =
null;
    String hsukubunga = null, hjangkawaktu = null;
    try {
        ◦ stm.executeUpdate("TRUNCATE
nasabah_kategorikal");
    } catch (SQLException ex) {
        Logger.getLogger(FKategorikal.class.getName()).log(Level.SEVERE,
null, ex);
    }
    for (int i = 0; i < jumBaris; i++) {
        try {
            int no = Integer.valueOf(tblNasabah.getValueAt(i,
0).toString());
            String produk = tblNasabah.getValueAt(i, 1).toString();
            long plafond = Long.valueOf(tblNasabah.getValueAt(i,
2).toString());
            double sukubunga =
Double.valueOf(tblNasabah.getValueAt(i, 3).toString());
            double jangkawaktu =
Double.valueOf(tblNasabah.getValueAt(i, 4).toString());
```

```

        long omzet = Long.valueOf(tblNasabah.getValueAt(i,
5).toString());
        String jenis_usaha = tblNasabah.getValueAt(i,
6).toString();
        int kolektibilitas =
Integer.valueOf(tblNasabah.getValueAt(i, 7).toString());
        if (produk.contains("KI")) {
            hproduk = "KI";
        } else if (produk.contains("KMK")) {
            hproduk = "KMK";
        } else if (produk.contains("TL")) {
            hproduk = "TL";
        }
        if (plafond < 1665000000L) {
            hplafond = "Rendah";
        } else if (plafond >= 1665000000L && plafond <
3330000000L) {
            hplafond = "Standar";
        } else if (plafond >= 3330000000L && plafond <=
5000000000L) {
            hplafond = "Tinggi";
        }
        if (omzet < 999000000L) {
            homzet = "Rendah";
        } else if (omzet >= 999000000L && plafond <
1998000000L) {
            homzet = "Standar";
        } else if (plafond >= 1998000000L && plafond <=
3000000000L) {
            homzet = "Tinggi";
        }
        if (sukubunga < 13.75) {
            hsukubunga = "Rendah";
        } else {
            hsukubunga = "Tinggi";
        }

```

```

    }
    if (jangkawaktu < 24) {
        hjangkawaktu = "Pendek";
    } else {
        hjangkawaktu = "Panjang";
    }
    if (kolektibilitas == 1) {
        hkolek = "Lancar";
    } else {
        hkolek = "Macet";
    }
    //masukkan data ke database
    String query = "insert into
nasabah_kategorikal(no,plafond,suku_bunga,jangka_waktu,omzet,j
enis_usaha,produk,kolektibilitas) "
        + "values("+no+",","" + hplafond + "," + hsukubunga +
"," + hjangkawaktu + "," + homzet + "," + jenis_usaha + "," +
hproduk + "," + hkolek + ")";
    stm.executeUpdate(query);
    System.out.println((i+1)+" "+query);
} catch (SQLException ex) {
    Logger.getLogger(FHome.class.getName()).log(Level.SEVERE,
    null, ex);
}
}
connect.Close();
FKategorikal fkat=new FKategorikal();
fkat.setVisible(true);

```

➤ *Source Code Uji Data*

```

private int ujiData(String[] data){
    //data[0] = data plafond P
    //data[1] = data suku bunga R
    //data[2] = data jangka waktu T
    //data[3] = data omzet O

```

```

//data[4] = data jenis usaha B
//data[5] = data produk S
//data[6] = data kolektibilitas K
String query;
String query1;
ResultSet rs;
String hasil = null;
try {
    //6 parameter
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and T='"+data[2]+" and O='"+data[3]+" and
B='"+data[4]+" and S='"+data[5]+"";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values('" +data[7]+"','" + data[0] + "','" + data[1] + "','" +
data[2] + "','" + data[3] + "','" + data[4] + "','" + data[5] + "','" + data[6] +
"', '" + hasil + "','" + sama + "')";
        stm.executeUpdate(query1);
        return 1;
    }

    //5 parameter
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and T='"+data[2]+" and O='"+data[3]+" and
B='"+data[4]+" and S is NULL";
    rs = stm.executeQuery(query);

```



```

if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where P="+data[0]+" and
R="+data[1]+" and T="+data[2]+" and O="+data[3]+" and
S="+data[5]+" and B is NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

```

```

    }
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and T='"+data[2]+" and B='"+data[4]+" and
S='"+data[5]+" and O is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values(" +data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and O='"+data[3]+" and B='"+data[4]+" and
S='"+data[5]+" and T is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```

```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+"' and
T='"+data[2]+"' and O='"+data[3]+"' and B='"+data[4]+"' and
S='"+data[5]+"' and R is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R='"+data[1]+"' and
T='"+data[2]+"' and O='"+data[3]+"' and B='"+data[4]+"' and
S='"+data[5]+"' and P is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```

```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

//4 parameter
query = "select * from rule where P="+data[0]+" and
R="+data[1]+" and T="+data[2]+" and O="+data[3]+" and B is NULL
and S is NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

```

```

        query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and T='"+data[2]+" and B='"+data[4]+" and O is NULL
and S is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and T='"+data[2]+" and S='"+data[5]+" and B is NULL
and O is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```

```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and O='"+data[3]+" and B='"+data[4]+" and T is NULL
and S is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and O='"+data[3]+" and S='"+data[5]+" and B is NULL
and T is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```

```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
    query = "select * from rule where P="+data[0]+" and
R="+data[1]+" and B="+data[4]+" and S="+data[5]+" and T is NULL
and O is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P="+data[0]+" and
T="+data[2]+" and O="+data[3]+" and B="+data[4]+" and R is NULL
and S is NULL";
    rs = stm.executeQuery(query);

```

```

        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where P='"+data[0]+" and
T='"+data[2]+" and O='"+data[3]+" and S='"+data[5]+" and B is NULL
and R is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;

```



```

    }
    query = "select * from rule where P='"+data[0]+" and
T='"+data[2]+" and B='"+data[4]+" and S='"+data[5]+" and R is NULL
and O is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
O='"+data[3]+" and B='"+data[4]+" and S='"+data[5]+" and T is NULL
and R is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```

```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R='"+data[1]+"' and
T='"+data[2]+"' and O='"+data[3]+"' and B='"+data[4]+"' and P is NULL
and S is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R='"+data[1]+"' and
T='"+data[2]+"' and O='"+data[3]+"' and S='"+data[5]+"' and B is NULL
and P is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```

```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
    query = "select * from rule where R="+data[1]+" and
T="+data[2]+" and B="+data[4]+" and S="+data[5]+" and P is NULL
and O is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R="+data[1]+" and
O="+data[3]+" and B="+data[4]+" and S="+data[5]+" and P is NULL
and T is NULL";
    rs = stm.executeQuery(query);

```

```

if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where T="+data[2]+" and
O="+data[3]+" and B="+data[4]+" and S="+data[5]+" and P is NULL
and R is NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

```

```

    }

    //3 parameter
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and T='"+data[2]+" and B is NULL and S is NULL and O
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
R='"+data[1]+" and O='"+data[3]+" and B is NULL and S is NULL and T
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
    }

```

```

        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P="+data[0]+" and
R="+data[1]+" and B="+data[4]+" and T is NULL and S is NULL and O
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P="+data[0]+" and
R="+data[1]+" and S="+data[5]+" and B is NULL and T is NULL and O
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());

```

```

        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
T='"+data[2]+" and O='"+data[3]+" and B is NULL and S is NULL and R
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }

```

```

        query = "select * from rule where P='"+data[0]+" and
T='"+data[2]+" and B='"+data[4]+" and R is NULL and S is NULL and O
is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values('" + data[7] + "','" + data[0] + "','" + data[1] + "','" +
data[2] + "','" + data[3] + "','" + data[4] + "','" + data[5] + "','" + data[6] +
"', '" + hasil + "','" + sama + "')";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where P='"+data[0]+" and
T='"+data[2]+" and S='"+data[5]+" and B is NULL and R is NULL and O
is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```



```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+"' and
O='"+data[3]+"' and B='"+data[4]+"' and R is NULL and S is NULL and T
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+"' and
O='"+data[3]+"' and S='"+data[5]+"' and B is NULL and R is NULL and T
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```

```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
    query = "select * from rule where P="+data[0]+" and
B="+data[4]+" and S="+data[5]+" and R is NULL and T is NULL and O
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R="+data[1]+" and
T="+data[2]+" and O="+data[3]+" and B is NULL and S is NULL and P
is NULL";
    rs = stm.executeQuery(query);

```

```

        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where R="+data[1]+" and
T="+data[2]+" and B="+data[4]+" and P is NULL and S is NULL and O
is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;

```

```

    }
    query = "select * from rule where R='"+data[1]+" and
T='"+data[2]+" and S='"+data[5]+" and B is NULL and P is NULL and O
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values(" + data[7] + "," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R='"+data[1]+" and
O='"+data[3]+" and B='"+data[4]+" and T is NULL and S is NULL and P
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```

```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R="+data[1]+" and
O="+data[3]+" and S="+data[5]+" and B is NULL and T is NULL and P
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R="+data[1]+" and
B="+data[4]+" and S="+data[5]+" and P is NULL and T is NULL and O
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```

```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where T='"+data[2]+" and
O='"+data[3]+" and B='"+data[4]+" and R is NULL and S is NULL and P
is NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where T='"+data[2]+" and
O='"+data[3]+" and S='"+data[5]+" and B is NULL and R is NULL and P
is NULL";
rs = stm.executeQuery(query);

```

```

if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where T='"+data[2]+" and
B='"+data[4]+" and S='"+data[5]+" and P is NULL and R is NULL and O
is NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

```

```

    }
    query = "select * from rule where O='"+data[3]+'"' and
B="'+data[4]+'"' and S="'+data[5]+'"' and P is NULL and R is NULL and T
is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values('"+data[7]+'','" + data[0] + "','" + data[1] + "','" +
data[2] + "','" + data[3] + "','" + data[4] + "','" + data[5] + "','" + data[6] +
"', '" + hasil + "','" + sama + "')";
        stm.executeUpdate(query1);
        return 1;
    }

//2 parameter
    query = "select * from rule where P='"+data[0]+'"' and
R="'+data[1]+'"' and T is NULL and O is NULL and B is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
    }

```



```

        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P="+data[0]+" and
T="+data[2]+" and R is NULL and O is NULL and B is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P="+data[0]+" and
O="+data[3]+" and T is NULL and B is NULL and R is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());

```

```

        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where P='"+data[0]+" and
B='"+data[4]+" and T is NULL and O is NULL and R is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }

```

```

        query = "select * from rule where P='"+data[0]+" and
S='"+data[5]+" and T is NULL and O is NULL and B is NULL and R is
NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where R='"+data[1]+" and
T='"+data[2]+" and P is NULL and O is NULL and B is NULL and S is
NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```

```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R='"+data[1]+"' and
O='"+data[3]+"' and T is NULL and P is NULL and B is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where R='"+data[1]+"' and
B='"+data[4]+"' and T is NULL and O is NULL and P is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```

```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
    query = "select * from rule where R="+data[1]+" and
S="+data[5]+" and T is NULL and O is NULL and B is NULL and P is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where T="+data[2]+" and
O="+data[3]+" and P is NULL and R is NULL and B is NULL and S is
NULL";
    rs = stm.executeQuery(query);

```

```

if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where T='"+data[2]+"' and
B='"+data[4]+"' and P is NULL and O is NULL and R is NULL and S is
NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

```

```

    }
    query = "select * from rule where T='"+data[2]+" and
S='"+data[5]+" and P is NULL and O is NULL and B is NULL and R is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where O='"+data[3]+" and
B='"+data[4]+" and T is NULL and P is NULL and R is NULL and S is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "

```

```

        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where O='"+data[3]+" and
S='"+data[5]+" and T is NULL and R is NULL and B is NULL and P is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where B='"+data[4]+" and
S='"+data[5]+" and T is NULL and O is NULL and R is NULL and P is
NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {

```



```

        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}

//1 elemen
query = "select * from rule where P="+data[0]+" and R is NULL
and T is NULL and O is NULL and B is NULL and S is NULL";
rs = stm.executeQuery(query);
if(rs.next()){
    System.out.println(query+" => "+rs.getRow());
    char sama = 'n';
    hasil=rs.getString("K");
    if (hasil.equals(data[6])) {
        sama = 'y';
    }
    query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
        + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
    stm.executeUpdate(query1);
    return 1;
}
query = "select * from rule where R="+data[1]+" and P is NULL
and T is NULL and O is NULL and B is NULL and S is NULL";
rs = stm.executeQuery(query);

```

```

        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where T="+data[2]+" and P is NULL
and R is NULL and O is NULL and B is NULL and S is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }

```

```

        query = "select * from rule where O='"+data[3]+'"' and P is NULL
and R is NULL and T is NULL and B is NULL and S is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
            stm.executeUpdate(query1);
            return 1;
        }
        query = "select * from rule where B='"+data[4]+'"' and P is NULL
and R is NULL and T is NULL and O is NULL and S is NULL";
        rs = stm.executeQuery(query);
        if(rs.next()){
            System.out.println(query+" => "+rs.getRow());
            char sama = 'n';
            hasil=rs.getString("K");
            if (hasil.equals(data[6])) {
                sama = 'y';
            }
            query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
                + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";

```

```

        stm.executeUpdate(query1);
        return 1;
    }
    query = "select * from rule where S='"+data[5]+'"' and P is NULL
and R is NULL and T is NULL and O is NULL and B is NULL";
    rs = stm.executeQuery(query);
    if(rs.next()){
        System.out.println(query+" => "+rs.getRow());
        char sama = 'n';
        hasil=rs.getString("K");
        if (hasil.equals(data[6])) {
            sama = 'y';
        }
        query1 = "insert into
hasil(no,plafond,suku_bunga,jangka_waktu,omzet,jenis_usaha,produk,k
olektibilitas,hasil,sama) "
            + "values("+data[7]+"," + data[0] + "," + data[1] + "," +
data[2] + "," + data[3] + "," + data[4] + "," + data[5] + "," + data[6] +
"," + hasil + "," + sama + ")";
        stm.executeUpdate(query1);
        return 1;
    }
} catch (SQLException ex) {

Logger.getLogger(FKategorikal.class.getName()).log(Level.SEVERE,
null, ex);
}
return 0;
}

```

➤ ***Source Code Matriks Discernibility***

```

private void bentukMatrixDiscernibility() {
    int jumBaris =
tblNasabahKategorikal.getRowCount(); //total data

```

```

        jumDataTraining = (int) Math.ceil(persenTraining *
        jumBaris);
        ArrayList[][] matrix = new
        ArrayList[jumDataTraining][jumDataTraining]; //bentuk
        matrix discernibility ukuran n x n
        for (int i = 0; i < jumDataTraining; i++) {
            matrix[i][i] = new ArrayList();
            String koleki =
            tblNasabahKategorikal.getValueAt(i, 7).toString();
            //kolektibilitas [K]
            for (int j = (i + 1); j < jumDataTraining; j++) {
                matrix[i][j] = new ArrayList();
                matrix[j][i] = new ArrayList();
                String kolekj =
                tblNasabahKategorikal.getValueAt(j, 7).toString();
                //kolektibilitas [K]
                if (!koleki.equals(kolekj)) {
                    String plafondi =
                    tblNasabahKategorikal.getValueAt(i, 1).toString();
                    //plafond [P]
                    String sukubungai =
                    tblNasabahKategorikal.getValueAt(i, 2).toString(); //suku
                    bunga [R]
                    String jangkawaktui =
                    tblNasabahKategorikal.getValueAt(i, 3).toString();
                    //jangka waktu [T]
                    String omzeti =
                    tblNasabahKategorikal.getValueAt(i, 4).toString();
                    //omzet [O]

```

```

        String jenissusahai =
tblNasabahKategorikal.getValueAt(i, 5).toString(); //jenis
usaha [B]

        String produki =
tblNasabahKategorikal.getValueAt(i, 6).toString();
//produk [S]

        String plafondj =
tblNasabahKategorikal.getValueAt(j, 1).toString();
//plafond [P]

        String sukubungaj =
tblNasabahKategorikal.getValueAt(j, 2).toString(); //suku
bunga [R]

        String jangkawaktuj =
tblNasabahKategorikal.getValueAt(j, 3).toString();
//jangka waktu [T]

        String omzetj =
tblNasabahKategorikal.getValueAt(j, 4).toString();
//omzet [O]

        String jenissusahaj =
tblNasabahKategorikal.getValueAt(j, 5).toString(); //jenis
usaha [B]

        String produkj =
tblNasabahKategorikal.getValueAt(j, 6).toString();
//produk [S]

        ArrayList data = new ArrayList();
        if (!plafondi.equals(plafondj)) {
            data.add("P");
        }
        if (!sukubungai.equals(sukubungaj)) {
            data.add("R");
        }

```

```

        if (!jangkawaktui.equals(jangkawaktuj)) {
            data.add("T");
        }
        if (!omzeti.equals(omzetj)) {
            data.add("O");
        }
        if (!jenisusahai.equals(jenisusahaj)) {
            data.add("B");
        }
        if (!produki.equals(produkj)) {
            data.add("S");
        }
        matrix[i][j] = data;
        matrix[j][i] = data;
    }
}

ArrayList cRule[] = new ArrayList[jumDataTraining];
System.out.println("Sebelum:");
for (int i = 0; i < jumDataTraining; i++) {
    for (int j = 0; j < jumDataTraining; j++) {
        System.out.print(matrix[i][j].toString() + " ");
    }
    System.out.println();
}
//operasi subset per baris
for (int i = 0; i < jumDataTraining; i++) { //cek per
baris
    cRule[i] = new ArrayList();
    for (int j = 0; j < jumDataTraining; j++) {
        for (int k = (j + 1); k < jumDataTraining; k++) {

```

```

        if (matrix[i][j].size() < matrix[i][k].size()) {
            if (matrix[i][k].containsAll(matrix[i][j]) &&
!matrix[i][j].isEmpty() && !matrix[i][k].isEmpty()) {
                matrix[i][k] = new ArrayList();
            }
        } else {
            if (matrix[i][j].containsAll(matrix[i][k]) &&
!matrix[i][j].isEmpty() && !matrix[i][k].isEmpty()) {
                matrix[i][j] = new ArrayList();
            }
        }
    }
}

System.out.println("Sesudah:");
for (int i = 0; i < jumDataTraining; i++) {
    for (int j = 0; j < jumDataTraining; j++) {
        System.out.print(matrix[i][j].toString() + " ");
    }
    System.out.println();
}

for (int i = 0; i < jumDataTraining; i++) {
    ArrayList r = new ArrayList();
    for (int j = 0; j < jumDataTraining; j++) {
        if (!matrix[i][j].isEmpty()) {
            r.add(matrix[i][j]);
        }
    }
    cRule[i] = r;
}

for (int i = 0; i < cRule.length; i++) {

```



```
        System.out.println((i + 1) + ". " + cRule[i].toString()  
+ " dengan jumlah data=" + cRule[i].size());  
        pecahRule(cRule[i],i);  
    }  
}
```

DAFTAR PUSTAKA

- [1] Wahyuningtyas, Grizelda. (2014). “Aplikasi Data Mining Untuk Penilaian Kredit Menggunakan Metode Pohon Keputusan Fuzzy”.
- [2] Winda Aprianti dan Imam Mukhlash, ICoDSE (2014). “ The Application of Rough Set and Fuzzy Rough Set Based Algorithm to Classify Incomplete Meteorological Data”, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7062674.
- [3] The Internasional Arab Journal of Information Technology, vol. 7, no. 2, April (2010). “Credit Scoring Models Using Soft Computing Methods: A Survey”.
- [4] Claudia. C, Ciptohartono. (2014). “*Algoritma Klasifikasi Naïve Bayes untuk Menilai Kelayakan Kredit*”.
- [5] Y. Ping and L. Yongheng, vol. 38, no. 9, pp. 11300–11304, Sep. (2011) “Neighborhood rough set and SVM based hybrid credit scoring classifier,” Expert Systems with Applications.
- [6] Jiawei Han, Micheline Kamber dan Jian Pei (2012), 3rd Ed “Data Mining Concepts and Techniques”.
- [7] Imamatul Ummah dan Winda Aprianti “Rough Set(Decission Tree and Decision Rules)”.
- [8] Tessy Badriyah “Algoritma Klasifikasi: Decision Tree”, Politeknik Elektronik Negri Surabaya.
- [9] Setyawan, Indra. (2013). “Perancangan dan Implementasi Sistem Pakar Berbasis Android Untuk Membantu Pengambilan Keputusan Dalam Menentukan Alat Kontrasepsi Keluarga Berencana”. Institut Teknologi Sepuluh Nopember, Surabaya.
- [10] Dan A Simovici dan Chabane Djeraba (2008). “Mathematical Tools for Data Mining, Set Theory, Partial Orders, Combinatorics”.

- [11] Zdzislaw Pawlak, Journal Of Telecommunications And Information Technology (2002) “Rough Set Theory and Its Applications”.
- [12] Astuti, Hanim Maria, Iqbal, Mohammad, Mukhlash, Imam “Multiclass Classification for Meteorological Data using Modified CBS Algorithm with Multiple Minimum Support” Indian Journal of Science and Technology, *Vol 8(12), 70652, June 2015.*

BIODATA PENULIS



Anisa Sofia lahir di Surabaya, 03 Oktober 1991, merupakan putri kedua dari tiga bersaudara dari pasangan Bapak Ir. Suhadak dan Ibu Ir. Sugistiana Dwi Lusilowati. Penulis menghabiskan masa kecilnya di kota Surabaya dan menempuh pendidikan formal di SDN Ketintang III/569 Surabaya, SMPN 32 Surabaya, dan SMA Muhammadiyah 2 Surabaya. Pada tahun 2010, Penulis melanjutkan studi di Matematika ITS melalui jalur PMDK Mandiri. Di Jurusan Matematika ini Penulis terdaftar dengan NRP 1210 100 024. Penulis mengambil Bidang Minat Ilmu Komputer. Penulis juga aktif dalam kepanitiaan pada beberapa kegiatan yang diadakan HIMATIKA dan ITS. Untuk keterangan lebih jelas mengenai Tugas Akhir ini dapat menghubungi Penulis melalui email : anisasofiaah@gmail.com